



Tim Olimpiade Komputer Indonesia

Tantangan

Bebras Indonesia 2017

Bahan belajar Computational Thinking

Tingkat
SMP

Diterbitkan Oleh :

NBO Bebras Indonesia



<http://bebras.or.id>

Pengantar

Tantangan Bebras Indonesia adalah kompetisi yang dilaksanakan secara online dan serentak dengan memberikan soal-soal yang telah dipersiapkan dalam Workshop Bebras Internasional, pada periode bebras week di minggu kedua bulan November.

Tantangan Bebras Indonesia dibagi dalam 3 kategori, yaitu:

1. Siaga, untuk siswa SD dan yang sederajat
2. Penggalang, untuk siswa SMP dan yang sederajat
3. Penegak, untuk siswa SMA dan yang sederajat.

Untuk kategori Siaga (SD) diberikan 10 soal yang harus diselesaikan dalam waktu 40 menit. Untuk kategori Penggalang (SMP) dan Penagak (SMA) masing-masing diberikan 12 soal yang harus diselesaikan dalam waktu 45 menit.

Tantangan Bebras Indonesia 2017 dapat berjalan lancar berkat dukungan penuh dari GDP Labs yang menyediakan dan mengelola <https://olympia.id> sebagai sistem aplikasi untuk lomba online. Selain dari itu juga LAPI Divusi yang membantu mengelola situs <http://bebras.or.id>

Selain dari itu para Koordinator Bebras Biro dan tim yang tersebar di 33 perguruan tinggi di seluruh Indonesia yang langsung berhubungan dengan para siswa dalam menyelenggarakan Tantangan Bebras Indonesia 2017.

Penyiapan soal-soal dan pengelolaan Tantangan Bebras Indonesia 2017 dilaksanakan oleh Tim Olimpiade Komputer Indonesia (TOKI), yaitu: Inggriani (ITB), Adi Mulyanto (ITB), Suryana Setiawan (UI), Julio Adisantoso (IPB), Rully Soelaiman (ITS). Yudhi Purwananto (ITS), Yugo K. Isal (UI), dan Fauzan Joko Sularto (UPJ). Penyiapan soal juga dibantu oleh Mewati Ayub (UKM), Cecilia Nugraheni dan Vania Natalia (Unpar), serta penyiapan buku ini dibantu oleh Inez Perera, dan Rana R. Natawigena.

Bahan belajar Computational Thinking Tantangan Bebras Indonesia 2017 ini dibagi dalam tiga buku sesuai kategori, yaitu buku untuk Tingkat SD (Siaga), Tingkat SMP (Penggalang), dan Tingkat SMA (Penagak).

Karya ini dilisensikan di bawah lisensi Creative Commons Attribution-Non Commercial-No Derivatives 4.0 International (CC BY-NC-SA 4.0)

Hal ini berarti Anda bebas untuk menggunakan dan mendistribusikan buku ini, dengan ketentuan:

- Attribution: Apabila Anda menggunakan materi-materi pada buku ini, Anda harus memberikan kredit dengan mencantumkan sumber dari materi yang Anda gunakan.
- Non Commercial: Anda tidak boleh menggunakan materi ini untuk keperluan komersial, seperti menjual ulang buku ini.
- ShareAlike: Apabila Anda mengubah atau membuat turunan dari materi-materi pada buku ini, Anda harus menyebarkan kontribusi Anda di bawah lisensi yang sama dengan materi asli.

Computational Thinking

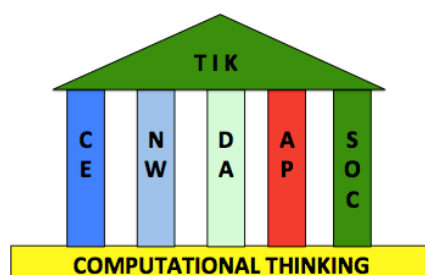
Kemampuan berpikir kreatif, kritis dan komunikasi serta kolaborasi adalah kemampuan yang paling penting dalam (*21st century learning*) pembelajaran di abad kedua-puluh-satu, di antara kemampuan-kemampuan lainnya seperti membaca, matematik, sains. Siswa zaman sekarang perlu untuk mengembangkan keterampilan berpikir, menguasai pengetahuan tentang konten dari persoalan yang dihadapi (*content knowledge*), dan mempunyai kompetensi sosial dan emosional untuk mengarungi kehidupan dan lingkungan kerja yang semakin kompleks.

Di bidang "*Computing*" (diterjemahkan ke dalam bahasa Indonesia menjadi "Informatika"), kemampuan berpikir yang perlu dikuasai sejak pendidikan dasar adalah "*Computational Thinking*" (CT). CT adalah proses berpikir untuk memformulasikan persoalan dan solusinya, sehingga solusi tersebut secara efektif dilaksanakan oleh sebuah agen pemroses informasi ("komputer", robot, atau manusia). CT adalah sebuah metoda dan proses berpikir untuk penyelesaian persoalan dengan menerapkan:

- Dekomposisi dan formulasi persoalan, sedemikian rupa sehingga dapat diselesaikan dengan cepat dan efisien serta optimal dengan menggunakan komputer sebagai alat bantu;
- Organisasi dan analisis data secara logis;
- Representasi data melalui abstraksi dalam bentuk model, dan melakukan simulasi;
- Otomasi solusi melalui berpikir algoritmik (langkah-langkah yang terurut);
- Identifikasi, analisis, dan implementasi solusi yang mungkin dengan tujuan langkah dan sumberdaya yang efisien dan efektif;
- Pengenalan pola persoalan, generalisasi serta mentransfer proses penyelesaian persoalan ke sekumpulan persoalan sejenis.

CT adalah sebuah cara berpikir untuk memecahkan persoalan, merancang sistem, memahami perilaku manusia. CT melandasi konsep informatika. Di dunia saat ini dimana komputer ada di mana-mana untuk membantu berbagai segi kehidupan, CT harus menjadi dasar bagaimana seseorang berpikir dan memahami dunia dengan persoalan-persoalannya yang semakin kompleks. CT berarti berpikir untuk menciptakan dan menggunakan beberapa tingkatan abstraksi, mulai memahami persoalan sehingga mengusulkan pemecahan solusi yang efektif, efisien, "fair" dan aman. CT berarti memahami konsekuensi dari skala persoalan dan kompleksitasnya, tak hanya demi efisiensi, tetapi juga untuk alasan ekonomis dan sosial.

Di negara maju, "*Computer Science*" (yang di Indonesia diterjemahkan sebagai "Informatika") sudah mulai diajarkan sejak usia dini di tingkat pendidikan dasar, dengan materi dan kegiatan yang dirancang dengan mengacu ke kerangka kurikulum yang disusun oleh persatuan guru-guru, asosiasi profesi informatika, perusahaan terkemuka di bidang informatika dan TIK, serta organisasi-organisasi nirlaba yang peduli terhadap perlunya edukasi tentang informatika sejak usia dini [<https://k12cs.org>]. Kerangka kurikulum Informatika tersebut mendefinisikan lima bidang pengetahuan yaitu: Sistem Komputer (CE), Jaringan Komputer (NW), Analisis Data (DA), Algoritma dan Pemrograman (AP), dan Aspek Sosial dari pemanfaatan Informatika (SOC). Selain pengetahuan, juga didefinisikan praktek-praktek komputasi untuk mengemas pengetahuan dan mempraktekkannya, yaitu: pembinaan menumbuhkan budaya komputasi, menciptakan artefak, berkolaborasi untuk mewujudkan suatu produk TIK, menguji dan memperbaiki/menyempurnakan artefak TIK, mengenali dan mendefinisikan problema-problema komputasi, berkomunikasi tentang komputasi, dan mengembangkan serta menggunakan abstraksi.



Gambar 1. Hubungan Computational Thinking, Informatika dan TIK

Bagaimana Belajar Computational Thinking?

Berpikir itu dapat dipelajari dan diasah dengan berlatih, serta mengkonstruksi pola pikir berdasarkan pengalaman. *Computational Thinking* juga dapat dipelajari dengan cara berlatih menyelesaikan persoalan-persoalan yang terkait komputasi, melalui persoalan sehari-hari. Lewat latihan-latihan yang menarik, siswa menerapkan teknik yang cocok (dekomposisi, abstraksi, pengenalan pola, representasi data, algoritmik) untuk mendapatkan solusi. Setelah latihan, siswa diharapkan melakukan refleksi serta mengkonstruksi pengetahuan berpikir, kemudian membentuk pola berpikir komputasional, yang semakin lama semakin tajam, cepat, efisien, dan optimal.

Apa perbedaan ICT/TIK (Teknologi Informasi dan Komunikasi) dengan Informatika?

Sejalan dengan itu, ICT (*Information and Communication Technology*, dalam bahasa Indonesia disebut Teknologi Informasi dan Komunikasi/TIK) mulai **dibedakan** dengan Informatika. TIK mengarah ke penggunaan teknologi dan perangkat/gadget, sedangkan Informatika mengarah ke keilmuan dan desain produk-produk informatika baik yang nyata (piranti pintar), maupun yang abstrak seperti program aplikasi, dan algoritma.

Kemampuan TIK lebih mengarah ke penggunaan teknologi dan perangkat/*gadget*, sedangkan Informatika mengarah ke keilmuan komputasinya. Penggunaan TIK yang dimaksud bukan hanya ketrampilan menggunakan gadget dan aplikasinya, tetapi juga kemampuan untuk menggunakan dan memanfaatkan konten dengan bijak.

Agar bangsa Indonesia mampu bersaing dengan negara lain, anak Indonesia tidak cukup menjadi pengguna teknologi saja, melainkan harus lebih kreatif dan inovatif untuk menciptakan produk-produk TIK. Untuk ini, siswa perlu mempelajari informatika.

Tantangan Bebras

(Bebras Computational Thinking Challenge)

Situs: <http://bebras.org>

Bebras challenge (semula adalah *Algorithmic Challenge* kemudian menjadi *Computational Thinking Challenge*), diinisiasi oleh Prof. Valentina Dagiene dari Lithuania sejak tahun 2004, adalah kompetisi yang diadakan tahunan bagi siswa berumur 5 s.d. 18 tahun dan sudah diikuti oleh sekitar 1.3 juta siswa yang berasal dari 50 negara. Komunitas Bebras sebagian besar adalah para pembina IOI seperti halnya Indonesia, adalah sekumpulan akademisi yang peduli ke pendidikan informatika bagi siswa sekolah dasar dan menengah.

Bebras mengikuti perkembangan CT, lewat “*challenge*” atau tantangan yang diberikan untuk *problem solving* terkait informatika untuk kehidupan sehari-hari, yang disajikan secara menarik dan lucu. Lewat Tantangan Bebras, siswa diajak “membangun” ketrampilan berpikir untuk menyelesaikan persoalan, yaitu melalui pendekatan *constructionism* yang diperkenalkan oleh Seimort Papert dari MIT. Siswa diajak belajar dengan mencoba menjawab tantangan. Jadi, tantangan Bebras bukan lomba sekedar untuk menang tetapi yang lebih penting adalah untuk belajar berpikir dan menyelesaikan persoalan. Kepada peserta yang meraih peringkat tinggi, akan diberikan sertifikat.

Tujuan Tantangan Bebras:

- Memotivasi siswa Untuk mulai tertarik ke topik-topik informatika dan memecahkan persoalan dengan menggunakan informatika
- Men-stimulasi minat siswa ke informatika
- Mendorong siswa untuk menggunakan “TIK” dengan lebih intensif dan kreatif dalam aktivitas belajarnya
- Menyemangati siswa untuk berpikir lebih dalam dari pada sekedar ke komputer/alatnya dan TIK.

Tantangan bebras diselenggarakan sekali setahun pada saat hampir bersamaan di seluruh dunia, sepanjang pekan Bebras, yang ditetapkan pada minggu pertama bulan November.

Bebras Indonesia

Situs: <http://bebras.or.id>

Bebras dikelola oleh pembina Pusat/Nasional TOKI. Indonesia mulai bergabung ke komunitas internasional bebras, dan untuk pertama kali mengadakan Tantangan Bebras dalam bahasa Indonesia pada tahun 2016. Tantangan diselenggarakan secara online digelar di 12 kota dengan 1553 peserta yang terdiri dari 193 peserta kategori Siaga (SD), 198 peserta kategori Penggalang (SMP), dan 968 peserta kategori Penegak (SMA/SMK/MAN) yang berasal dari 125 Sekolah (23 SD, 27 SMP, 75 SMA). Penyelenggaraan Tantangan Bebras dikoordinasi oleh Perguruan Tinggi yang menjadi Mitra bebras Indonesia, dan dapat diselenggarakan di Perguruan Tinggi Koordinator atau di sekolah. Peserta ada yang menggunakan komputer, tablet, bahkan handphone.

Bagaimana Berpartisipasi pada Tantangan Bebras 2018?

Pembina Bebras Indonesia bekerja sama dengan Perguruan Tinggi mitra dengan dukungan supporter. Perguruan Tinggi (diutamakan Program Studi Informatika dan Matematika) yang berminat untuk menjadi mitra Bebras akan dihubungkan dengan Perguruan Tinggi Pembina Utama TOKI, dan sekolah yang berminat untuk mengikut-sertakan siswa dapat menghubungi Perguruan Tinggi Mitra Bebras terdekat. Sebagai persiapan, Pembina Bebras tingkat Nasional juga bersedia menjadi narasumber untuk pelatihan dosen/guru yang akan akan bergabung.

Silahkan kontak via email ke info@bebras.or.id.

Contoh-contoh soal dan latihan online dalam bahasa Indonesia dapat diakses di <https://olympia.id>.

Untuk Latihan di arena Bebras Indonesia

Siapapun dapat berlatih secara mandiri di situs <https://olympia.id> dengan langkah:

1. Akses "Create New Account", atau dari: <https://olympia.id/login/signup.php>
2. Setelah mengisi data diri secara lengkap dan password benar, anda menekan "Create New Account", kemudian anda harus membuka email untuk konfirmasi.
3. Buka email anda
4. Klik link yang diberikan oleh Olympia ke email anda
5. Anda melakukan "Enroll" ke Bebras Challenge.

Negara-Negara Kontributor

Setiap soal di buku ini diberi bendera yang menandakan negara asal penyusun soal. Namun banyak pihak yang terlibat dalam mengedit, menerjemahkan, dan menyediakan material tambahan.

Bebras Indonesia berterima kasih kepada komunitas Bebras internasional karena memungkinkan kami untuk menggunakan soal-soal yang telah mereka kembangkan.

Bendera Negara Kontributor Soal-Soal pada Buku SMP Tantangan Bebras 2017

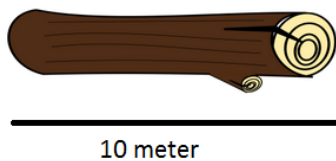


DAFTAR SOAL

- Membangun Bendungan (halaman 8)
- Senyum (halaman 9)
- Upah Membantu (halaman 10)
- Kode Kartu (halaman 12)
- Pedang dan Perisai (halaman 13)
- Kursi Musik (halaman 14)
- Pesta Ulang Tahun (halaman 16)
- Kiri Kanan (halaman 18)
- Lampu Merah Biru (halaman 19)
- Kota Penuh Putaran (halaman 20)
- Menari Sesuai Sorakan Penonton (halaman 22)
- Memindahkan Dadu (halaman 24)
- Area Parkir (halaman 26)
- Robot Pengumpul Permen (halaman 28)
- Kurung (halaman 30)
- Jembatan Honomakato (halaman 32)
- Majalah Dinding (halaman 33)
- Membangun Bendungan (halaman 35)
- Barisan Anjing (halaman 37)
- Robot (halaman 38)



Bebras Jordan menebang pohon dan memotongnya sehingga setiap pohon menghasilkan batang pohon yang panjangnya 10 meter.



Bebras Ali ingin membangun bendungan dan membutuhkan kayu sepanjang 4 meter sebanyak 7 batang, dan kayu sepanjang 3 meter sebanyak 7 batang. Ali dapat memotong batang pohon yang sudah dipotong Jordan sesuai dengan keperluannya. Jordan ingin memberikan batang pohon kepada Ali dengan jumlah sesedikit mungkin.

Tantangan:

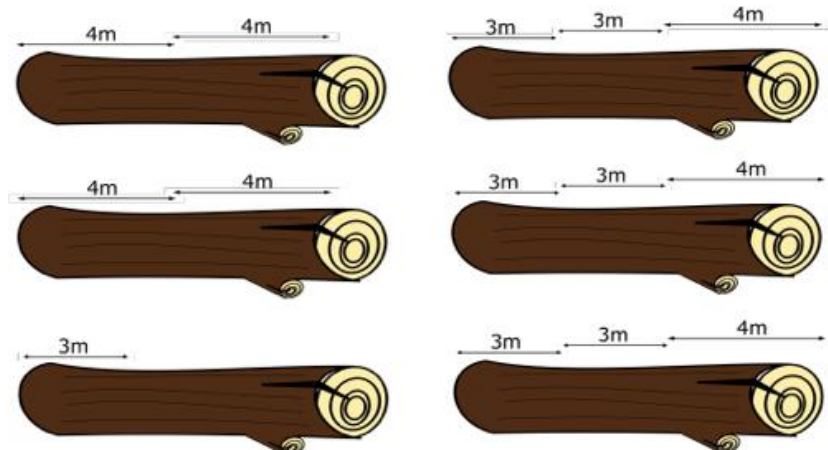
Berapa banyaknya batang pohon minimal yang dapat dipakai Ali untuk membangun bendungan?

Pilihan Jawaban:

- 5
- 6
- 7
- 8

Jawaban:

Jawaban yang tepat adalah 6.



Ini Informatika!

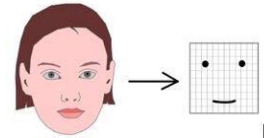
Masalah “memotong-material” (“*cutting-stock*” problem) adalah persoalan untuk memotong material yang panjangnya standar seperti pipa, atau batang kayu menjadi potongan-potongan material berbagai ukuran, dengan sesedikit mungkin material yang dipotong. Ini adalah contoh persoalan optimasi dalam industri. Persoalannya dapat diformulasikan sebagai persoalan pemrograman linier.

https://en.wikipedia.org/wiki/Cutting_stock_problem

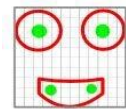


Berang-berang menemukan suatu alat untuk mendeteksi wajah tersenyum dengan kamera. Alat tersebut bekerja dengan 2 langkah:

Tahap-1: transformasi foto wajah menjadi semacam "smiley" yang diberi dua titik dan garis yang menunjukkan posisi mata dan mulut.

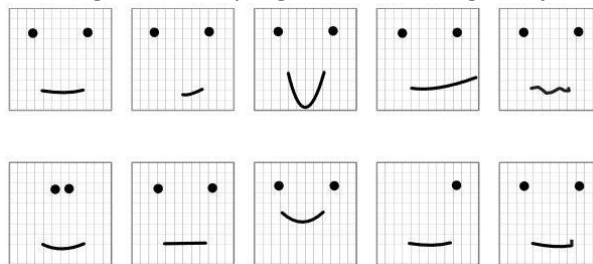


Tahap-2: deteksi apakah wajah tersenyum dengan mencocokkan gambar hasil tahap-1 dengan pola yang terdiri dari garis merah dan titik hijau. Wajah dideteksi sebagai wajah tersenyum, jika dan hanya jika gambarnya menyentuh semua titik hijau dan tidak menyentuh garis merah.



Tantangan:

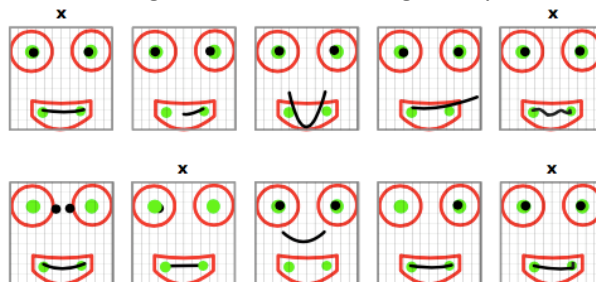
Berapa gambar hasil tahap-1 sebagai berikut, yang dideteksi sebagai wajah tersenyum?



Jawaban:

Jawaban yang benar adalah 4.

Penjelasan: Gambar yang ditandai dengan x terdeteksi sebagai senyum



Ini Informatika!

Ilmuwan/Ahli Informatika mengembangkan sistem yang dapat mengenali sesuatu (misalnya wajah manusia) pada foto, atau kamera yang selalu hidup untuk menangkap citra. Teknologi ini sekarang mengadopsi konsep yang disebut mesin pembelajar, atau bahkan *deep learning*. Teknologi ini juga memasukkan tahap pra pemrosesan citra, memodelkan, dan memakai aturan seperti yang diberikan contohnya pada tantangan ini.

Adalah sangat penting bahwa pendekatan yang digunakan dalam latihan ini sangat disederhanakan, dan mempunyai beberapa kekurangan. Misalnya, empat wajah yang diberikan sebagai alternatif jawaban, hanya dua yang secara tepat merepresentasi senyum. Anda harus mengingat baik-baik, bahwa kelak, rancangan sistem pengenalan citra yang sebenarnya, tidak menghasilkan seperti ini.

- [Robot passport checker rejects asian man's photo having closed eyes](#)
- [Research shows smartphone fingerprint scanners aren't as secure as we think they are](#)



Untuk mengisi liburannya selama 6 hari, Ana dan Bobi merencanakan tinggal di desa nenek. Kebetulan, di sana ada tiga petani A, B, dan C yang membutuhkan bantuan untuk menggarap sawahnya masing-masing. Mereka menawarkan Ana dan Bobi upah jika mau membantu mereka. Masing-masing petani tersebut memberikan penawaran yang berbeda:

- Petani A menawarkan 10 ribu rupiah buat masing-masing (Ana dan Bobi) setiap hari.
- Petani B hanya akan memberi Bobi sepuluh ribu rupiah pada hari pertama kemudian setiap berikutnya menaikkan sebesar 10 ribu menjadi 20 ribu, 30 ribu, dan seterusnya, sementara ia akan memberi Ana di hari pertama 100 ribu rupiah dan kemudian diturunkan 10 ribu rupiah setiap hari berikutnya menjadi 90 ribu, 80 ribu, dan seterusnya.
- Petani C tidak tertarik dibantu Bobi, sehingga ia hanya akan memberi 1 ribu rupiah di hari pertama saja dan tidak akan memberi apapun di hari berikutnya. Sementara untuk Ana, ia akan memberikan seribu rupiah pada hari pertama, lalu setiap hari berikutnya dua kali lipat sebelumnya. Jadi Ana akan mendapatkan seribu rupiah, 2 ribu rupiah, 4 ribu rupiah, 8 ribu rupiah dan seterusnya.

Mereka berniat untuk melewati setiap hari masa liburannya di desa nenek dengan membantu petani, dan mereka berdua sudah berjanji untuk bekerja pada petani yang sama. Mengenai upah, mereka juga diam-diam sudah sepakat untuk membagi sama rata dari yang diperoleh berdua.

Tantangan:

Kepada petani yang mana mereka bekerja sehingga mendapat upah yang paling banyak ?

Pilihan Jawaban:

- A
- B
- C
- A atau B

Jawaban:

Jawaban yang tepat adalah B.

Perhitungan upah petani A sangat sederhana. Masing-masing mendapat upah 50 ribu per hari, sehingga masing-masing akan memperoleh 500 ribu rupiah selama 10 hari.

Petani B lebih rumit perhitungannya:

- Bobi mendapat $10+20+30+40+50+60+70+80+90+100$
- Anna mendapat $100+90+80+70+60+50+40+30+20+10$

Bagaimana mendapatkan hasil penjumlahannya? Pada hari pertama, keduanya memperoleh 110 rupiah, demikian juga 110 rupiah pada hari kedua ... Secara keseluruhan, selama 10 hari keduanya akan memperoleh 1100 rupiah, sehingga masing-masing akan mendapat 550. Hore!

Petani C lebih sulit lagi. Ini hasil penjumlahan upah Ana dan Bobi:

$$1 + 1 + 2 + 4 + 8 + 16 + 32 + 64 + 128 + 256 + 512 =$$

$$2 + 2 + 4 + 8 + 16 + 32 + 64 + 128 + 256 + 512 =$$

$$4 + 4 + 8 + 16 + 32 + 64 + 128 + 256 + 512 =$$

$$8 + 8 + 16 \dots \text{wah ada pola,}$$

Mari abaikan sisanya,

$$\text{Akan berakhir dengan: } 512 + 512 = 1024$$

Jika mereka membagi 1024 untuk mereka berdua, setiap anak akan memperoleh 512.

Maka, jawabannya adalah Petani B

Ini Informatika!

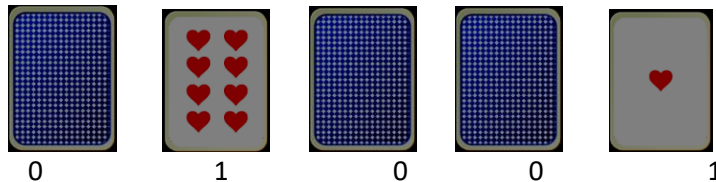
Ilmuwan/Ahli informatika seringkali melakukan perhitungan seperti yang dilakukan untuk pemecahan persoalan ini. Misalnya jika ingin menghitung waktu eksekusi dari sebuah algoritma (upah adalah analogi dari waktu eksekusi). Upah yang diperoleh adalah waktu eksekusi dari operasi yang diperlukan untuk suatu algoritma.

Selain itu, perhatikan betapa efisien menjumlahkan upah petani B. Menjumlahkan semua angka, jumlahkan secara paralel dari depan dan belakang pada saat yang sama, dan membagi dua

Pada kasus ini, kasus terakhir menguatirkan, karena jumlah operasi berlipat dua setiap hari, yang mencerminkan dua kali jika data bertambah. Kalau Ana dan Bobi mempunyai waktu 1 hari lebih, mereka berdua akan mendapatkan upah dengan upah Bobo ditambah kelipatan dari upah.



Terdapat lima kartu yang diletakkan berderet. Setiap kartu berisikan lambang hati dalam jumlah tertentu, dari kiri ke kanan secara berurutan adalah 16, 8, 4, 2, dan 1 hati. Di bagian bawah kartu terdapat tulisan angka 0 atau 1. Angka 1 dituliskan jika kartu di atasnya terbuka (dan terlihat gambar hatinya!). Angka 0 dituliskan jika kartu tertutup. Kartu-kartu dapat digunakan untuk membangkitkan kode-kode untuk angka-angka. Sebagai contoh, terdapat 9 gambar hati yang dapat terlihat pada gambar berikut ini, sehingga kode untuk angka 9 adalah 01001.



Tantangan:

Temukan kode untuk 26 gambar hati.

Jawaban:

Jawaban yang benar adalah 11010.

Ini Informatika!

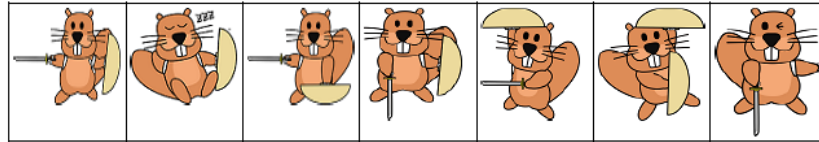
Hati ditempatkan di kartu sehingga hanya ada satu hati pada kartu sebelah kanan, dan setiap kartu di sebelah kiri sesudahnya mengandung dua kali hati. Maka, akan ada 1, 2, 4, 8, dan 16 hati saat dibaca dari kiri ke kanan.

Dapat dibuktikan bahwa setiap angka dari 0 s.d. 31 dapat direpresentasikan dengan kombinasi tertentu menggunakan kartu tsb. Kita juga dapat melakukan perhitungan dengan menggunakan kartu-kartu ini.

Pengaturan dari hati dipakai dalam sistem bilangan biner yang hanya menggunakan 0 dan 1, yang dipakai komputer untuk menyimpan informasi. Kita juga dapat mengkode teks, gambar, video, musik dengan hanya menggunakan bilangan 0 dan 1.




Taro sedang bermain pedang dan perisai dengan 7 teman-temannya dan mem-foto-nya. Diagram berikut ini menunjukkan foto-foto posisi kesukaan mereka masing-masing.



Mereka ingin memiliki foto-foto tersebut yang telah disusun menjadi satu gambar bersama, agar dapat dipasang di halaman sekolah. Dalam susunannya, setiap pedang harus menunjuk pada berang-berang yang lain, dan setiap perisai harus menghalangi pedang yang ditunjukkan kepadanya. Nah, Taro sudah menempati suatu posisi I di pojok kanan bawah pada dalam gambar tersebut.

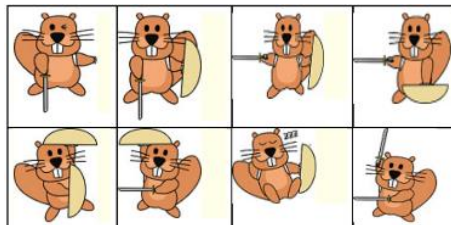
Tantangan:

Padankanlah foto-foto ke-7 teman-teman Lucia (bilangan 1 sampai dengan 7) dengan ruang-ruang yang masih kosong (huruf A, B, ..., G) agar susunan yang diharapkan.

A	B	C	D
E	F	G	I 

Jawaban:

Jawaban yang benar adalah:



Ini Informatika!

Ini sebenarnya adalah teka-teki yang sangat rumit. Hanya beberapa gambar yang mengarah pada pencarian yang sangat memakan waktu di antara semua solusi yang mungkin (tetapi tidak benar). Jika Anda menambahkan hanya satu gambar ke teka-teki 6 buah, Anda akan memiliki 6 kali lebih banyak kemungkinan berbeda untuk menempatkan 7 kartu di tempat-tempat kosong. Untuk n kartu, Anda memiliki $(n-1)! = 1 * 2 * 3 * \dots * (n-2) * (n-1)$ kemungkinan solusi yang berbeda. Jadi dalam hal ini ada 720 kemungkinan solusi yang berbeda (tetapi hampir semuanya salah).

Namun, dengan menggunakan pemikiran logis, ruang pencarian dapat dipangkas oleh banyak hal. Misalnya, semua berang-berang dengan tongkat menunjuk ke bawah harus ditempatkan di baris atas, dan hanya ada satu berang-berang yang dapat ditempatkan tepat di atas Lucia.

Pencarian lengkap dapat dilakukan dengan menggunakan algoritma yang disebut backtracking. Ketika menggunakan algoritma backtracking, ruang pencarian bisa menjadi sangat besar. Inilah sebabnya mengapa pemangkasan itu penting.



Kelompok berjumlah 7 berang-berang memainkan "kursi-musik" yaitu berpindah kursi saat musik dimainkan.



Saat musik dimulai, setiap berang-berang harus berpindah ke kursi searah dengan putaran jarum jam. Satu kursi dapat ditempati oleh lebih dari satu berang-berang.

Pada setiap putaran, Berang-berang besar (A) akan berpindah tiga (3) kursi berlawanan arah jarum jam. Berang-berang sedang (C dan D) akan berpindah dua (2) kursi berlawanan arah jarum jam, sedangkan Berang-berang kecil (E, G, dan F) hanya akan berpindah satu (1) kursi searah jarum jam.

Tantangan:

Jika pada awalnya posisi masing-masing sebagaimana terlihat pada gambar di atas, kursi mana yang TIDAK diduduki berang-berangnya tepat setelah putaran ke-3?

Pilihan Jawaban:

- 2 dan 7
- 3 dan 7
- 1 dan 2
- 1 dan 4

Jawaban:

Jawaban yang benar adalah 2 dan 7.

Berang-berang besar pindah 3 kursi melawan arah jarum jam.

Jadi A akan berpindah dari kursi 1 ke 6.

B berpindah dari kursi 6 ke kursi 4.

Berang-berang sedang pindah 2 kursi melawan arah jarum jam.

C berpindah dari kursi 4 ke 5.

D berpindah dari kursi 3 ke kursi 4.

Berang-berang kecil pindah 1 kursi searah jarum jam.

E akan pindah dari kursi 2 ke kursi 5, F dari kursi 7 ke 3, dan G dari kursi 5 ke 1.

Jadi setelah putaran ke-3, kursi 1, 3, dan 6 akan ditempati oleh 1 berang-berang di setiap kursi, sedangkan kursi 4 dan 5 akan ditempati oleh 2 berang-berang di setiap kursi.

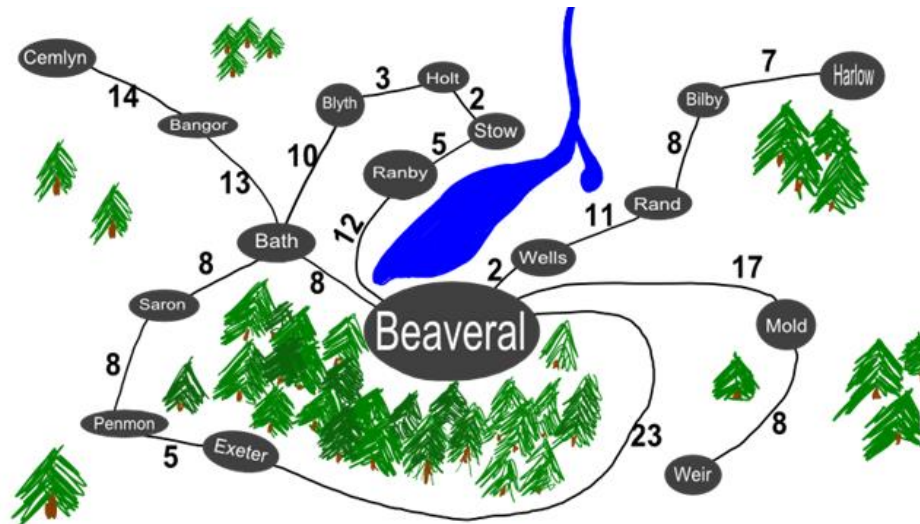
Hanya 2 kursi yang kosong, yaitu kursi 2 dan 7.

Ini Informatika!

Salah satu cara untuk memecahkan masalah ini adalah dengan mengingat bahwa semua gerakan setiap berang-berang tetap sama sepanjang putaran. Penguraian informasi tersebut memungkinkan kita untuk fokus pada masalah mereka tanpa terganggu oleh pengulangan yang menyertainya. Dengan melakukan ini, Anda dapat menyederhanakan algoritma dan mengidentifikasi polanya.



Patra tinggal di desa Beaveral. Ia ingin mengundang teman-temannya untuk merayakan ulangtahunnya. Namun ia hanya mengundang teman-teman yang tinggal tidak lebih dari 20 km dari rumahnya agar tak kemalaman pulang. Jarak rumah teman-temannya dimunculkan dalam angka pada peta ssebagai berikut.



Tantangan:

Teman-teman yang diundang, dan rumahnya berjarak lebih dari 17 akan disediakan jemputan. Tentukan teman-teman Patra yang diundang dan akan dijemput.

Pilihan Jawaban:

- Ranby
- Stow
- Mold
- Blyth
- Saron
- Holt
- Rand

Jawaban:

Jarak paling pendek dari Beaveral ke setiap desa lainnya dapat diperoleh dengan menjumlahkan jalur antara desa-desa yang dilalui, dan menentukan yang minimal. Hanya desa dengan jarak lebih kecil dari 20 km yang akan dicatat.

Urutan mulai dari paling dekat sampai terjauh dicatat untuk meyakinkan bahwa semua sudah diperiksa. Inilah cara algoritma Dijkstra's bekerja:

Wells = 2 Bath = 8 Ranby = 12 Rand = 13 = 2 + 11 (Bilby=13+8=21 terlalu jauh lewat jalan ini) Saron = 16 = 8 + 8 (Penmon=16+8=24 terlalu jauh lewat jalan ini) Mold = 17 (Weir=17+8=25 terlalu jauh lewat jalan ini) Stow = 17 = 12 + 5 Blyth = 18 = 8 + 10 (Holt=18+3=21 terlalu jauh lewat jalan ini) Holt = 19 = 17 + 2

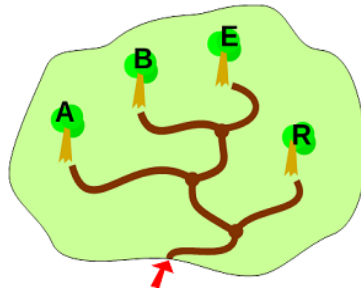
Cara mengikuti jalan ini sesuai dengan algoritma Dijkstra, yang dijelaskan secara sederhana pada video sebagai berikut: <https://www.youtube.com/watch?v=aBym6p-coic>. Ini tambahan kecil, karena kita punya batasan jarak saat kita mengeksplorasi graf.

Ini Informatika!

Informatika adalah ilmu pengetahuan yang berkaitan dengan penyimpanan dan representasi informasi. Pada tantangan ini, sebuah peta merepresentasikan sebaran desa dan jarak di antaranya. Persoalan ini bukan hanya tentang matematika sebab membutuhkan komputasi dan membandingkan, yang merupakan bagian dari informatika. Teknik yang sama dapat dipakai untuk struktur yang lebih kompleks, misalnya peta. Selain menghitung jarak tempuh, harus memperhatikan desa tetangga yang memenuhi persyaratan tertentu. Tantangan ini menguji anda apakah dapat mempunyai pandangan benar mengenai struktur data ini, dan dapat memanfaatkan dengan data yang disimpan di dalam struktur data tsb.

Penjelasan lebih lanjut tentang “*breadth-first search*”, salah satu teknik untuk meneliti jalur dalam struktur data graf, dapat ditemukan pada:

<https://www.khanacademy.org/computing/computer-science/algorithms/breadth-first-search/a/breadth-first-search-and-its-uses>



Berang-berang menciptakan sistem pengkodean kata yang disebut kode berang-berang, dengan memakai peta di atas:

- Setiap pohon di taman diberi nama dengan satu huruf.
- Kode untuk setiap huruf ditemukan dengan cara mencapai pohon tersebut dengan berbelok kiri (L) dan kanan (R).
- Kode untuk setiap huruf selalu dimulai dari pintu masuk taman (bertanda panah).

Contoh-contoh :

- Contoh 1: Kode untuk A adalah LL karena untuk mencapai pohon A dari pintu masuk taman kamu harus berbelok kiri dua kali.
- Contoh 2: Kode untuk kata BAR adalah LRLRLR.

Tantangan:

Berapa banyak huruf dalam kode berang-berang tersebut untuk kata BEAR? (Tuliskan angkanya!)

Jawaban:

Jawaban yang benar adalah 9.

Tabel berikut ini berisi kode kiri/kanan untuk semua huruf:

B	E	A	R
LRL	LRR	LL	R

Sehingga BEAR akan dikode menjadi LRLRRLLR yang terdiri dari 9 huruf.

Ini Informatika!

Jika suatu komputer mengganti huruf L dalam kode kiri/kanan dengan 0 dan huruf R dengan 1, maka kode kiri/kanan menjadi apa yang disebut kode biner. Peta untuk taman menjadi suatu struktur data komputer yang disebut pohon biner.

Hal ini berarti rute yang panjang dan rumit dapat disimpan oleh komputer dengan sangat mudah menggunakan ruang yang sangat kecil. Hal menarik mengenai kode ini adalah bahwa tidak diperlukan koma atau pemisah.

Cobalah mengubah kode (decode) jawaban untuk melihat bahwa kamu tidak perlu spasi untuk menunjukkan kode untuk setiap huruf cukup dan mulai dengan kode baru. Jenis kode seperti ini disebut prefix code. Hal ini berarti kode menjadi lebih singkat.



Berang-berang Beta mengirimkan pesan kepada temannya menggunakan 7 lampu, setiap lampu bisa berwarna merah atau biru. Beta menggunakan 5 lampu pertama untuk menunjukkan huruf yang dikirimkan. Untuk menunjukkan bahwa pesan itu benar, Beta memakai 2 lampu sisanya sebagai berikut:

- Lampu ke 6 berwarna biru jika jumlah lampu biru di antara lima lampu pertama adalah genap; jika tidak maka lampu ke 6 berwarna merah.
- Lampu ke 7 berwarna merah jika jumlah lampu merah di antara enam lampu pertama adalah genap; jika tidak maka lampu ke 7 berwarna biru.

Ingatlah, 0 adalah bilangan genap. Sebagai contoh, jika Beta sudah menentukan lima lampu pertama sebagai:



Maka lampu ke 6 harus berwarna biru (karena ada 2 lampu biru di antara 5 lampu pertama), dan lampu ke 7 harus berwarna merah (karena terdapat 3 lampu merah di antara 6 lampu pertama). Jadi, dia harus memakai pola berikut ini:



Tantangan:

Manakah pola berikut ini yang merupakan pesan yang benar?

Pilihan Jawaban:

- a)
- b)
- c)
- d)

Jawaban:



Jawaban yang benar adalah a)

Terdapat 1 lampu biru di antara 5 lampu pertama, jadi lampu ke 6 harus berwarna merah. Dengan demikian terdapat 5 lampu merah diantara 6 pertama, jadi lampu ke-7 harus berwarna biru. Untuk jawaban yang lainnya:

- Untuk (A), lampu ke 7 seharusnya biru, karena terdapat 1 lampu merah diantara enam lampu pertama;
- Untuk (B), lampu ke 6 seharusnya biru, karena terdapat 0 lampu biru diantara lima lampu pertama dan 0 adalah genap;
- Untuk (D), lampu ke 7 seharusnya biru, karena terdapat 3 lampu biru diantara enam lampu pertama.

Ini Informatika!

Kesalahan dapat terjadi ketika informasi dikirim. Misalnya, jika seseorang sedang menyebutkan nomor telepon, nomor kartu kredit, dll, selalu ada kemungkinan bahwa salah satu angka ditulis kembali secara tidak tepat atau urutannya salah. Beberapa kesalahan dapat dideteksi, dan kadang-kadang diperbaiki memakai teknik yang melibatkan *check digits*.

Pada tantangan ini, lampu ke 6 dan ke 7 berfungsi sebagai tanda untuk meyakinkan bahwa semua lampu sudah dikonfigurasi dengan tepat: jika warna lampu ke 6 atau ke 7 salah ditentukan, kita tahu bahwa pesan tersebut tidak benar.

Nomor Kartu kredit, nomor ISBN buku, atau kode UPC pada produk yang dibeli di toko, semuanya dilengkapi *check digits*, yang memberikan petunjuk jika terjadi kesalahan dalam pengiriman informasi.

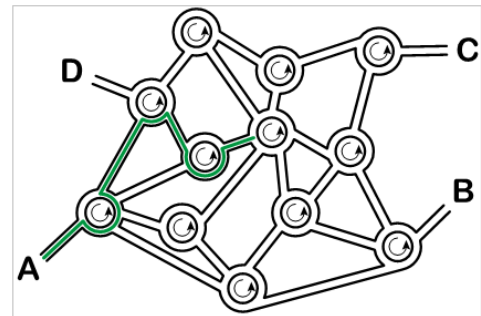


Suatu robot bekerja di suatu kota. Robot berjalan menyusuri jalan dari suatu tempat ke tempat lain mengikuti petunjuk yang diberikan. Pada setiap pertemuan beberapa ruas terdapat putaran (jalan memutar) dan robot akan memutar berlawanan arah jarum jam dan mengambil simpangan urutan tertentu sesuai yang petunjuk yang diberikan.

Petunjuk yang diberikan berupa angka-angka urutan simpangan. Misalnya petunjuk "4 1 2" akan diikuti robot sebagai berikut:

- Pada putaran pertama, ambil simpangan ke-4.
- Pada putaran kedua, ambil simpangan ke-1.
- Pada simpangan ketiga, ambil simpangan ke-2.

Jika robot mula-mula berada di A, maka petunjuk itu akan dijalaninya sebagai tergambar berikut ini dengan menyusuri jalan yang diberi warna gelap sebagai berikut:



Tantangan:

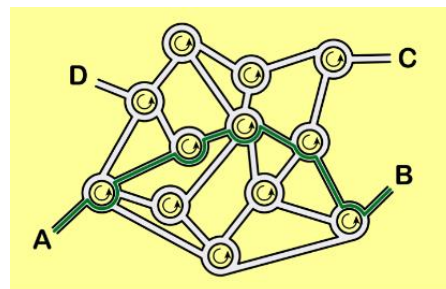
Jika pada awalnya robot berada di A, dan mengikuti petunjuk "3 1 3 2 3", dimanakah ia akan berakhir?

Pilihan Jawaban:

- A
- B
- C
- D

Jawaban:

Jawaban yang paling tepat adalah B.



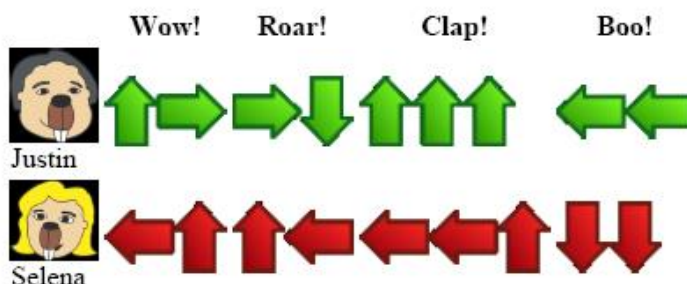
Ini Informatika!

Tantangan ini mengenalkan kita kepada salah satu elemen penting dalam bahasa pemrograman komputer, komposisi instruksi sekuensial (berturutan), yang berarti mengikuti instruksi satu per satu berdasarkan urutannya.

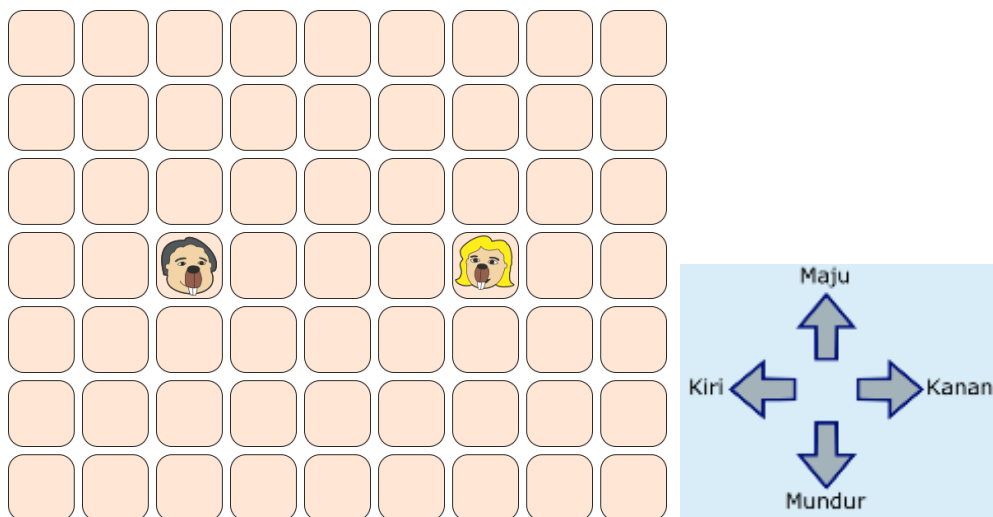
Urutan instruksi yang kita berikan ke robot ini adalah dasar dari program komputer. Dalam tantangan ini kita hanya diberikan sekumpulan instruksi yang sangat sederhana, atau dengan kata lain bahasa pemrograman yang sangat sederhana. Mungkin dalam situasi lain, bahasa sederhana seperti ini tidak cukup.



Dua berang-berang bernama Justin dan Selena, mempunyai gerakan tarian yang menyesuaikan jenis sorakan penonton. Mereka melakukan gerakan secara bersamaan satu kotak pada setiap saat sesuai arah anak panah di bawah ini. Tabel berikut ini menunjukkan bagaimana kedua berang-berang akan bergerak:



Sebagai contoh, jika penonton bersorak "Roar!" Justin akan bergerak satu kotak ke kanan kemudian satu kotak mundur ke belakang. Pada saat yang sama, Selena akan maju satu kotak ke depan dan satu kotak ke kiri. Mereka mulai dengan posisi di lantai dansa sebagai berikut:



Tantangan:

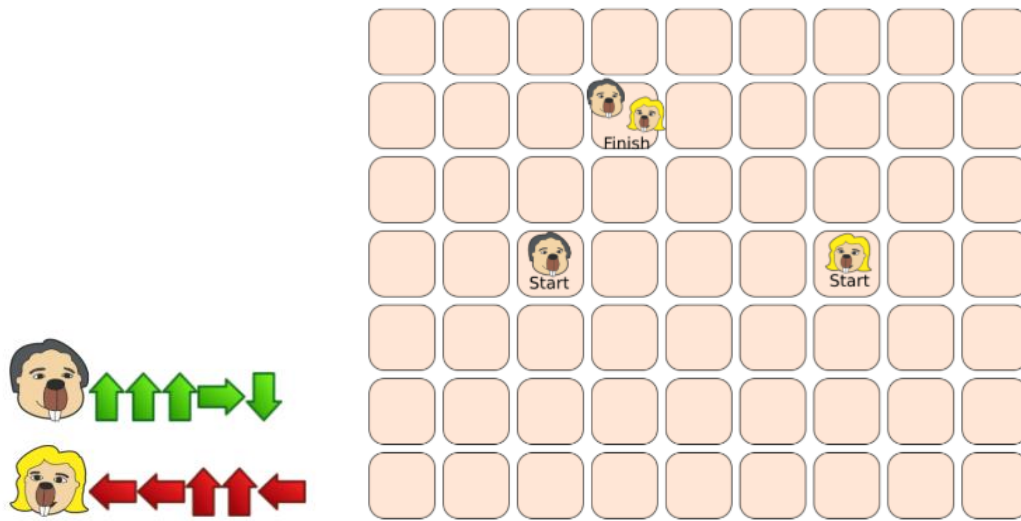
Manakah dari urutan sorakan berikut yang menyebabkan keduanya berhenti pada kotak yang sama?

Pilihan Jawaban:

- a) Boo! Roar!
- b) Clap! Roar!
- c) Wow! Roar!
- d) Roar! Roar!

Jawaban:

Jawaban yang benar adalah b) Clap! Roar!



Ini Informatika!

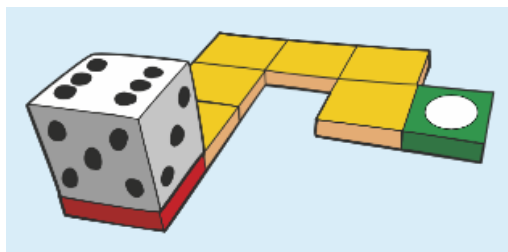
Persoalan ini memberikan suatu contoh tentang *parallel processing* atau *parallel computing*.

Dalam hal ini, setiap penari beraksi masing-masing secara independen. Ketika dua prosesor bertabrakan (yang dimaksud: mereka mencoba mengakses lokasi yang sama dalam kode atau memori), maka diperlukan kesepakatan prosesor mana yang mempunyai akses pertama terhadap bagian kode tersebut. Hal ini dapat dilakukan memakai misalnya *semaphore*. Dengan cara ini satu prosesor mempunyai kunci (*lock*) pada kode atau variabel yang sedang dipakai dan prosesor lainnya harus menunggu sampai prosesor pertama selesai menggunakan kode tersebut.

Ketika melakukan gerakan tari pada kertas, anda sebenarnya melakukan simulasi dari kedua penari. Simulasi penting dalam banyak situasi di dunia nyata. Misalnya, pada saat membuat suatu video game online, seringkali sulit untuk memperoleh akses ke sejuta pemain untuk menguji kode anda. Pemrogram biasanya akan mensimulasikan permainan dengan satu juta instans untuk melihat apakah permainan itu stabil atau akan menemui tabrakan.



Jack si berang-berang menggulirkan sebuah dadu sepanjang jalan tanpa pengeseran. Untuk memindahkan dadu dari satu petak ke petak berikutnya, Jack memutar dadu sepanjang pinggir yang ada di perbatasan antara dua petak. Dia melakukannya 7 kali sampai dadu mencapai petak berisi bulatan putih di sebelah kanan.



Perhatikan bahwa banyaknya titik di sisi kebalikan sebuah dadu selalu 7 (1 berlawanan dengan 6; 2 berlawanan dengan 5; 3 berlawanan dengan 4). Pada mulanya, sisi dengan 1 titik (berlawanan dengan sisi 6) ada di dasar dadu, seperti ditunjukkan pada gambar. Setelah memutar dadu sekali ke petak kedua, sisi dengan 2 titik (berlawanan dengan 5) akan berada di dasar dadu.

Tantangan:

Sisi dadu dengan berapa titik ada di dasar dadu saat dadu mencapai petak hijau di ujung?

Pilihan jawaban:

- a) 4
- b) 5
- c) 2
- d) 6

Jawaban:

Jawaban yang benar adalah a) 4.

Salah satu cara untuk memecahkan tantangan ini adalah dengan mengamati posisi semua sisi dadu saat dadu itu berputar.

Tapi sebaliknya, kita dapat mengurangi kompleksitasnya dengan mengamati 3 sisi saja, karena 3 sisi yang lainnya bisa ditentukan berdasarkan sisi yang berlawanan dengannya.

Misalnya, pertama-tama kita dapat mencatat posisi awal dadu dengan sisi 3, 5, dan 6 yang berada di posisi kanan, depan dan atas. Kita terus mengamati posisi ketiga sisi ini saat dadu berputar, selangkah demi selangkah sampai ke petak terakhir. Setelah dadu selesai bergerak, kita bisa tahu sisi mana yang berada di bawah.

		Posisi ●●●	Posisi ●●●●●	Posisi ●●●●●●
Langkah	Arah bergerak	Kanan	Depan	Atas
1	↑	Kanan	Atas	Belakang
2	↑	Kanan	Belakang	Bawah
3	↑	Kanan	Bawah	Depan
4	→	Bawah	Kiri	Depan
5	→	Kiri	Atas	Depan
6	↓	Kiri	Depan	Bawah
7	→	Atas	Depan	Kiri

Jadi sisi yang menghadap ke bawah di petak terakhir adalah 4, karena sisi 3 berada di sebelah atas.

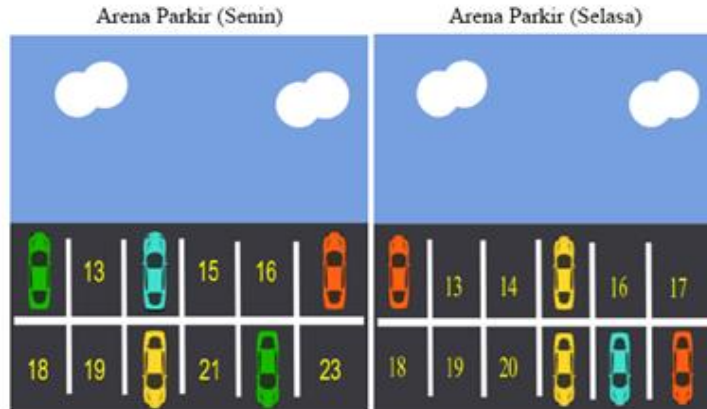
Ini Informatika!

Ini adalah contoh pemecahan masalah dengan mengamati hanya sebagian informasi di setiap tahap. Metode ini dapat mengurangi kerumitan dan penggunaan memori. Komputer sering diprogram seperti ini.

Salah satu alternatif dari metode ini adalah melacak jumlah titik di bagian bawah setelah setiap langkah. Tetapi jauh lebih sulit. Untuk beberapa putaran pertama, itu masih mudah. Tetapi setelah beberapa langkah, kita akan mengalami kesulitan karena beberapa angka tidak dapat dilihat. Selanjutnya, jika kita memprogram dua metode ini di komputer, kita akan melihat bahwa metode pertama, seperti yang ditunjukkan dalam jawaban, lebih cepat.



Terdapat 12 tempat untuk parkir mobil di area parkir. Setiap tempat diberi nomor. Gambar di bawah ini menunjukkan kondisi area parkir pada hari Senin dan pada hari Selasa.



Mobil yang parkir (pada hari Senin maupun Selasa) pasti akan parkir di tempat yang tergambar.

Tantangan:

Berapa banyak tempat parkir yang tidak pernah terisi mobil pada hari Senin maupun hari Selasa?

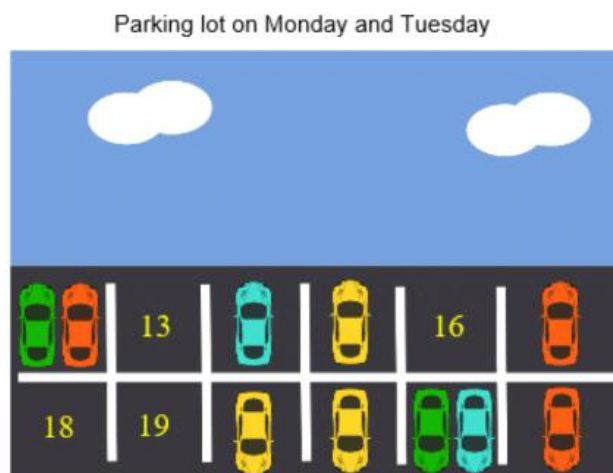
Pilihan Jawaban:

- 3
- 5
- 4
- 6

Jawaban:

Jawaban yang tepat adalah 4.

Kita dapat melihat tempat mana yang dipakai dengan menempatkan mobil-mobil dari kedua hari di arena parkir secara bersamaan.



Kemudian kita dapat menghitung banyaknya tempat kosong untuk menentukan 4 tempat kosong pada hari Senin dan Selasa.

Ini Informatika!

Semua data dapat dipandang sebagai rangkaian nol dan satu. Setiap nol atau satu disebut sebagai bit dan rangkaian bit disebut kode biner, representasi biner atau bilangan biner.

Pada kasus sini, kita dapat memodelkan tempat bersisi mobil sebagai satu (1) dan tempat kosong sebagai nol (0); sehingga satu tempat parkir direpresentasikan dengan satu bit. Kita mendapatkan rangkaian bit jika kita memandang bahwa tempat parkir berurutan. Misalnya kita mulai dari baris atas kemudian baru baris bawah untuk memperoleh 101001001010 untuk arena parkir pada hari Senin dan 100100000111 untuk arena parkir pada hari Selasa.

Soal ini meminta anda untuk menentukan pasangan bit mana dari kedua belas posisi pada kedua representasi biner yang keduanya bernilai nol (0).

Dalam logika, pernyataan: mencari tempat yang kosong pada (hari Senin dan hari Selasa) dapat dinegasi menjadi tempat yang isi (pada hari Senin atau hari Selasa)

Dalam hal ini, kita menentukan bit mana saja bernilai 1, dan operasi logik yang disebut OR. Perhatikan bagaimana kita dapat menghitung jawaban yang benar dengan melihat bahwa 101001001010 OR 100100000111 memberikan 10110100111. Ini memberikan hasil bilangan biner yang mempunyai 4 buah nol di dalamnya. Jadi, ada 4 tempat kosong pada hari Senin maupun juga kosong pada hari Selasa.

Robot Candy diprogram untuk mengumpulkan permen sebanyak mungkin yang terhampar di lantai yang terdiri dari petak-petak. Tugas tersebut dilakukan pada saat robot berjalan melalui petak demi petak lantai. Setiap petak di lantai sebagai tergambar di bawah ini memiliki 0, 1, 2 atau 3 permen. Robot Candy mulai dari petak S (untuk start) di kiri bawah dan berakhir di petak F (untuk finish) di kanan atas. Namun, Robot Candy memiliki keterbatasan: setiap kali berpindah, dia hanya bisa berpindah dari satu petak ke petak berikutnya di sebelah kanannya atau di sebelah atasnya.



Tantangan

Mengingat Robot Candy akan mengumpulkan permen sebanyak-banyaknya, berapa banyak permen yang dikumpulkan oleh Robot Candy?

Pilihan jawaban:

- 10
- 12
- 14
- 16
- 13

Jawaban:

Jawaban yang benar adalah 14.

Salah satu pendekatannya adalah mengisi tabel "terbaik" dari permen yang bisa dikumpulkan mengisi "sapuan diagonal" dari tabel. Awalnya kita punya 0 permen, jadi kita bisa memikirkan tabel sebagai berikut:

2	0	1	1	F
1	2	0	2	3
2	2	0	2	1
3	1	0	2	0
0	0	1	3	0

Dimana elemen dicetak dengan huruf tebal adalah jumlah maksimal permen yang bisa kita capai di setiap sel. Dengan bergerak naik robot akan mendapatkan 3 permen, dan dengan bergerak ke kanan robot akan mendapatkan 0 permen, jadi kita bisa memperbarui tabel kita:

2	0	1	1	F
1	2	0	2	3
2	2	0	2	1
3	1	0	2	0
0	0	1	3	0

Perhatikan sel yang ada di sebelah kanan angka 3 yang dicetak tebal dan di atas angka 0 yang dicetak tebal. Berapakah jumlah maksimum permen yang bisa kita kumpulkan untuk sampai ke sel ini? Kita harus sampai ke sel ini setelah mengumpulkan 3, bukan 0, permen. Jadi, kita bisa berada di sel ini setelah mengumpulkan 4 permen.

2	0	1	1	F
1	2	0	2	3
2	2	0	2	1
3	4	0	2	0
0	0	1	3	0

Melanjutkan dengan cara ini, kita dapat melihat bahwa jumlah maksimal permen yang bisa kita kumpulkan di dalam sebuah sel adalah jumlah permen dalam sel tersebut ditambah dan jumlah permen yang terbesar antara sel di sebelah kiri atas sel di bawahnya. Secara matematis hal ini dapat dinyatakan dengan

$$v(i, 0) = 0$$

$$v(0, j) = 0 \quad v(i, j) = c(i, j) + \max\{v(i-1, j), v(i, j-1)\}$$

dimana $v(i, j)$ adalah jumlah maksimum permen yang dapat dikumpulkan di sel (i, j) , dan $c(i, j)$ adalah jumlah permen dalam grid pada sel (i, j) pada awalnya.

Jika kita ingin mengacu ke sel paling kiri dan paling bawah, kita perlu menambahkan kolom angka nol ke kiri dan deretan angka nol ke bagian bawah tabel. Dengan menerapkan hubungan ini, kita bisa mengisi sisa tabel sebagai berikut:

0	8	9	10	12	14
0	6	9	9	11	14
0	5	7	7	9	10
0	3	4	4	6	6
0	0	0	1	4	4
0	0	0	0	0	0

dan oleh karena itu, kita bisa mengumpulkan 14 permen di sel F.

Inilah Informatika!

Menentukan solusi "terbaik" di antara serangkaian solusi yang mungkin adalah masalah yang sulit dan bermanfaat. Untuk masalah pengumpulan permen ini, kita bisa mencoba semua jalur yang mungkin, yang merupakan solusi *brute force*.

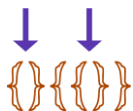
Sayangnya, ada banyak sekali jalur: khususnya, untuk masalah ini, ada 70 jalur yang berbeda. Dalam kasus khusus ini, kita dapat mencoba untuk menemukan beberapa bagian tugas yang "layak" dan mencoba (dan berhasil) untuk menemukan solusi terbaik dari sana. Karena kotaknya relatif kecil, kita bisa mengurangi kemungkinan lain yang harus lebih buruk lagi.

Solusi yang lebih efisien melibatkan pengisian tabel, seperti yang dijelaskan di atas, dengan menggunakan teknik yang disebut memoisasi dari *dynamic programming recurrence*. Artinya, setelah mendapatkan formula hubungan untuk solusi "terbaik" dari sel saat ini, berdasarkan sel ke kiri atau sel di bawah, kita melakukan 25 perhitungan, dalam hal ini, untuk menghitung jumlah maksimum dari permen tersedia, kita membangun solusi yang lebih besar dari kondisi awal.

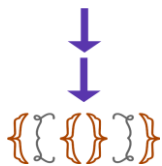
Sebuah toko permata membuat gelang yang tersusun atas bentuk-bentuk hiasan yang mirip tanda kurung. Ada dua bentuk yang berbeda: **P-1** dan **P-2**. Satu pasangan bentuk adalah **dua bentuk yang sama** yang diposisikan berhadapan: satu menghadap ke kanan (seperti tanda kurung buka) dan satu berikutnya menghadap ke kiri (Seperti tanda kurung tutup). Ada dua kemungkinan pasangan seperti pada gambar berikut, dan selanjutnya pasangan tersebut dinamakan **pasangan P-1** dan **pasangan P-2**).



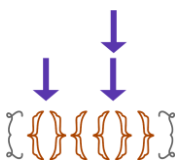
Proses pembuatan gelang dimulai dari sebuah pasangan apapun (P-1 atau P-2). Pasangan demi pasangan berikutnya disisipkan secara bersamaan pada suatu titik penyisipan: di kiri, atau di tengah, atau di kanan, dari suatu pasangan yang sudah ada pada gelang. Untuk memperjelasnya berikut ini ada sejumlah contoh.



Contoh ini dimulai dengan pasangan P-1. Pasangan P-1 kedua disisipkan di antara yang pertama, dan pasangan P-1 ketiga disisipkan di sebelah kiri dari P-1 yang pertama.



Contoh kedua dimulai dari pasangan P-1, kemudian disisipkan sebuah pasangan P-2 di tengah pasangan P-1 yang pertama, terakhir disisipkan pasangan P-1 di tengah pasangan P-2.



Contoh ketiga dimulai dari pasangan P-2. Pasangan P-1 pertama disisipkan di tengahnya, dan pasangan P-1 kedua disisipkan sebelah kiri pasangan P-1 pertama, tapi masih di dalam P-2. Terakhir pasangan P-1 ketiga disisipkan di tengah pasangan P-1 kedua.

Tantangan:

Manakah dari gelang-gelang berikut ini yang dibuat dengan cara yang telah diuraikan di atas?

Pilihan Jawaban:



Jawaban: Jawaban yang tepat adalah d)



Gelang mulai dibuat dengan dua kurung, menaruh sepasang di antaranya, kemudian menaruh pasangan lain di antara pasangan yang kedua. Semua gelang yang lain tidak dibentuk berdasarkan cara yang telah dijelaskan:

- A. Posisi 3 salah: sisi kanan dari hiasan 2 ditempatkan sebelum sisi kanan dari hiasan 1.
- B. Posisi 1 salah: dimulai dengan sisi kanan hiasan, seharusnya dari sisi kiri.
- C. Posisi 2 salah: susunan terdiri dari 3 sisi kiri dari satu hiasan dan kemudian 3 sisi kanan dari hiasan yang lain, jadi tidak disusun secara berpasangan.

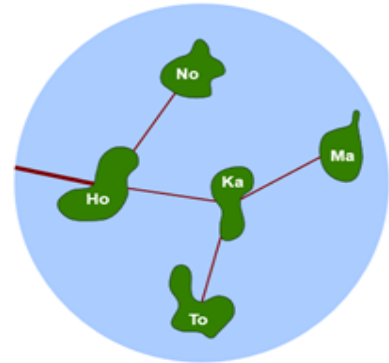
Ini Informatika!

Aturan untuk membuat gelang yang dijelaskan pada tantangan ini sama persis seperti aturan menyusun ekspresi kurung. Ilmuwan/Ahli Informatika menyebut ekspresi yang benar dengan "formula-benar" (*"well formed formula"*). Ekspresi yang mengandung kesalahan disebut "formula-salah". Suatu ekspresi yang merupakan "formula-benar" juga disebut "benar secara sintaks", yang berarti ekspresi taat pada aturan sintaks (aturan tata bahasa) yang ditentukan. Walaupun kesalahan sintaks mungkin akan sulit dikenali pada sebuah ekspresi yang rumit, kesalahan sintaks secara umum lebih mudah ditemukan dibandingkan "kesalahan semantik", yaitu kesalahan logika yang dibuat oleh pemrogram.



Kepulauan Honomakato terdiri dari lima pulau yang indah yaitu Ho, No, Ma, Ka, dan To. Pulau terbesar yaitu Ho terhubung ke daratan (pulau utama) oleh sebuah jembatan besar terbuat dari besi yang kokoh. Selain itu, terdapat jembatan-jembatan kayu yang menghubungkan Ho dengan No, Ho dengan Ka, Ka dengan Ma, dan Ka dengan To.

Penduduk kepulauan Honomakato berharap dibangun dua jembatan kayu lagi agar jika salah satu jembatan kayu putus, setiap pulau tetap masih bisa mencapai daratan lewat pulau Ho, karena jembatan dari Ho ke daratan tak mungkin rusak.



Tantangan:

Jembatan yang mana?

Pilihan Jawaban:

- Ho dengan To, dan No dengan Ma.
- Ka dengan No, dan No dengan Ma.
- Ho dengan To, dan Ma dengan To.
- Dua jembatan kayu tidak cukup untuk itu.

Jawaban:

Jawaban yang benar adalah a) Ho dengan To, dan No dengan Ma.

Kalau ada jembatan Ho-To, jika Ho-Ka atau Ka-To terputus, To masih terhubung. Selain itu, Ka masih terhubung juga melalui To kalau jembatan Ho-Ka putus.

Kalau ada jembatan No dan Ma:

- Ma masih terhubung melalui No, jika jembatan Ho-Ka atau Ka-Ma terputus.
- No terhubung dengan Ka dan Ma, jika jembatan Ho-No terputus.
- Ka masih terhubung melalui No dan Ma, jika jembatan Ho-Ka putus.

Jembatan Ma-To tidak dapat menghubungkan No kalau jembatan Ho-No terputus.

Jembatan Ka-No tidak dapat menghubungkan To jika jembatan Ho-Ka atau Ka-To putus.

Ini Informatik!

Jaringan kabel yang dibuat untuk kepulauan Honomakato menggambarkan hanya sebagian kecil dari seluruh internet. Strukturnya menunjukkan contoh dari semua jaringan internat secara global: komputer, HP, televisi, CCTV, dan apapun yang terkoneksi ke Internet saat ini, dapat digambarkan sebagai jejaring dari simpul, seperti pulau-pulau pada jaringan Honomakato.

Pada tahun 1960, motivasi utama dari penemuan internet adalah untuk membangun sebuah jejaring yang dengan cepat pulih jika ada masalah ("resilien"). Khususnya, tidak berfungsinya sebuah simpul tidak menyebabkan seluruh jaringan menjadi lumpuh (tak berfungsi). Jenis jejaring selain internet, misalnya jejaring lalu lintas antar kota, pengiriman logistik, juga sama pentingnya agar tidak mempunyai satu simpul yang jika lumpuh dapat menyebabkan semua jaringan lumpuh.

Ilmuwan/Ahli informatika menggunakan teori graf untuk menalarakan sebuah jaringan. Sebuah graf adalah sebuah jaringan terdiri dari simpul dan penghubung (jalur) antar simpul. Sebuah graf disebut "terhubung", jika setiap pasangan A dan B terhubung, yaitu dapat mencapai A dari B melalui satu atau lebih simpul. Jika ada sebuah simpul yang harus ada untuk membuat graf tersebut terhubung, maka simpul tersebut disebut "jembatan". Dalam sebuah jaringan yang cepat pulih terhubung kembali saat sebuah simpul gagal berfungsi, harus dihindari adanya simpul "jembatan". Untungnya, ada algoritma untuk mengenali simpul "jembatan". Robert Tarjan menemukan sebuah algoritma mendeteksi simpul jembatan yang efisien – dan juga banyak algoritma lainnya.



Sepuluh siswa sedang bekerja untuk menerbitkan majalah dinding sekolah. Setiap hari Jumat, mereka menulis dan memperbaiki artikel majalah dinding. Sebagian pekerjaan membutuhkan komputer. Pada tabel di bawah, sel yang berwarna menunjukkan bahwa seorang siswa membutuhkan komputer. Pada satu jam yang sama, satu komputer hanya dapat digunakan oleh seorang siswa.

		Jam						
		8:00	9:00	10:00	11:00	12:00	13:00	14:00
Siswa	1							
	2							
	3							
	4							
	5							
	6							
	7							
	8							
	9							
	10							

Tantangan:

Berapa jumlah minimum komputer yang dibutuhkan supaya mereka dapat bekerja sesuai dengan tabel di atas?

Pilihan Jawaban:

- a) 4
- b) 5
- c) 6
- d) 10

Jawaban:

Jawaban yang tepat adalah b) 5.

Pada jam 09:00 dan 10:00, 5 siswa memerlukan sebuah komputer. Berarti, paling sedikit kita membutuhkan 5 komputer supaya mereka semua bisa bekerja.

Kalau kita mengatur jadwalnya dengan benar, seperti di tabel berikut ini, 5 komputer akan cukup.

		Hours						
		8:00	9:00	10:00	11:00	12:00	13:00	14:00
Student	1		PC 3	PC 3				
	2			PC 1	PC 1	PC 1	PC 1	
	3	PC 1	PC 1					
	4					PC 3	PC 3	PC 3
	5		PC 4	PC 4				
	6				PC 2	PC 2		
	7			PC 5	PC 5	PC 5	PC 5	PC 5
	8		PC 5					
	9	PC 2	PC 2	PC 2				
	10						PC 2	PC 2

Ketika siswa datang untuk memakai komputer, mereka duduk di depan komputer yang pertama tersedia. Saat mereka selesai, siswa yang lain bisa datang dan duduk di komputer tersebut.

Ini Informatika!

Untuk memahami sejumlah besar data dan hubungan antara bermacam-macam tipe data, jalan yang terbaik adalah dengan menciptakan metode representasi data, contohnya dengan menggunakan tabel, grafik atau diagram.

Tergantung pada informasi yang dicari, kita bisa menunjukkan data yang sama secara berbeda. Dalam tantangan ini, kita bisa juga menunjukkan data yang sama dengan tabel lain seperti berikut:

Computer	Hours						
	8:00	9:00	10:00	11:00	12:00	13:00	14:00
PC 1	Student 3	Student 3	Student 2	Student 2	Student 2	Student 2	
PC 2	Student 9	Student 9	Student 9	Student 6	Student 6	Student 10	Student 10
PC 3		Student 1	Student 1		Student 4	Student 4	Student 4
PC 4		Student 5	Student 5				
PC 5		Student 8	Student 7	Student 7	Student 7	Student 7	Student 7

Penjadwalan dan penggunaan sumber daya yang optimal seringkali merupakan persoalan dalam kehidupan sehari-hari. Contoh: pemesanan kamar hotel, di mana satu kamar hanya boleh dipesan oleh satu pelanggan, dan alokasi kamar harus optimal, sebuah jadwal dalam bentuk tabel pemakaian yang optimal akan membantu.

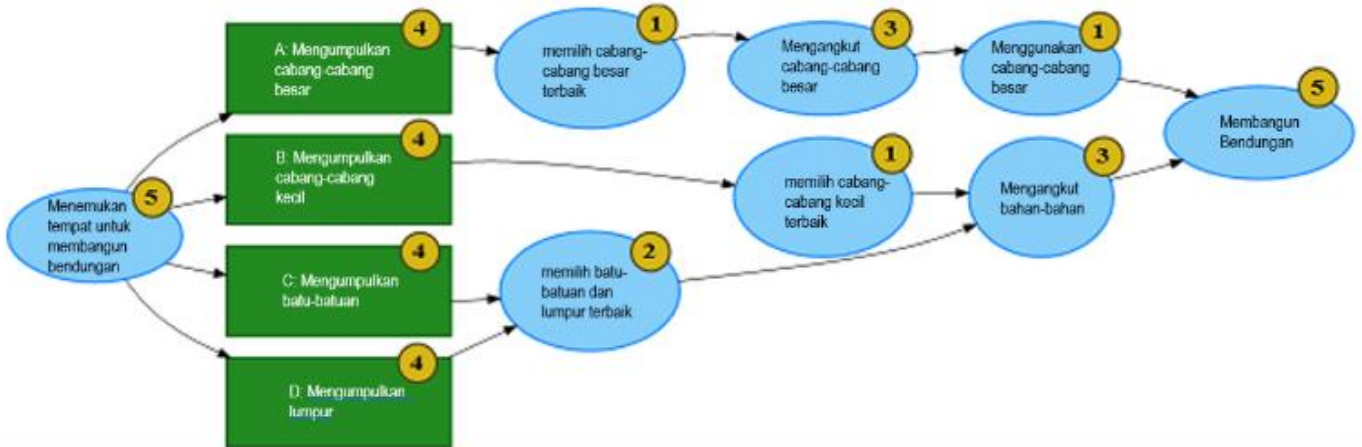
Cara lain untuk merepresentasi data adalah dalam bentuk graf, di mana simpul adalah interval, dan di antara dua simpul ada jalur yang merepresentasi bahwa kedua simpul beririsan (ada interseksi). Graf ini disebut graf interval. Sebuah pewarnaan graf interval yang optimal menyatakan penugasan/alokasi pemakaian sumber daya, yang memenuhi semua permintaan dengan sumberdaya yang paling sedikit” (Wikipedia)

[https://en.wikipedia.org/wiki/Table_\(information\)](https://en.wikipedia.org/wiki/Table_(information))

https://en.wikipedia.org/wiki/Interval_graph



Empat berang-berang sedang membangun bendungan. Mereka merencanakan tugas seperti gambar di bawah ini. Urutan tugas yang harus dilakukan ditunjukkan dengan tanda anak panah dan boleh dimulai setelah tugas sebelumnya selesai. Bilangan di dalam lingkaran kuning menunjukkan berapa jam waktu yang diperlukan untuk menyelesaikan suatu tugas. Setiap tugas dilakukan oleh seekor berang-berang.



Setelah berang-berang menemukan tempat terbaik untuk membangun bendungan, seekor berang-berang memutuskan untuk tidur siang dahulu selama 4 jam, sehingga tugasnya akan tertunda selama 4 jam.

Tantangan:

Pilihlah satu dari empat tugas yang harus ditunda supaya waktu tunggu dalam penyelesaian bendungan menjadi minimal.

Pilihan Jawaban:

- Mengumpulkan cabang-cabang besar
- Mengumpulkan cabang-cabang kecil
- Mengumpulkan batu-batuan
- Mengumpulkan lumpur

Jawaban:

Jawaban yang paling benar adalah b) Mengumpulkan cabang-cabang kecil.

“Penundaan 4 jam pada tugas A, C atau D akan menunda proyek keseluruhan dengan 4 jam. Tugas “B – Mengumpulkan cabang kecil” mempunyai kelonggaran waktu (slack time) sebesar satu jam. Jika tugas ini ditunda selama 4 jam, maka waktu total untuk menyelesaikan proyek hanya akan bertambah 3 jam. “A. Mengumpulkan cabang besar”, “C. Mengumpulkan lumpur”, “D. Mengumpulkan batu-batuan” adalah tugas yang berada di jalur kritis dari proyek. Hal ini berarti setiap penundaan tidak peduli berapa panjangnya akan mempengaruhi waktu total yang diperlukan untuk proyek. Jawaban benar adalah “B. Mengumpulkan cabang kecil.” bukanlah bagian dari jalur kritis. Hal ini berarti terdapat slack time untuk tugas ini, dalam hal ini slack time adalah satu jam.

Ini informatika!

Pemrosesan tugas-tugas, dengan memperhatikan kebergantungannya, adalah salah satu tipe masalah dalam informatika. Misalnya sistem operasi dari komputer modern memungkinkan beberapa program untuk berjalan secara simultan pada saat yang bersamaan. Sebaliknya, perangkat keras sering hanya

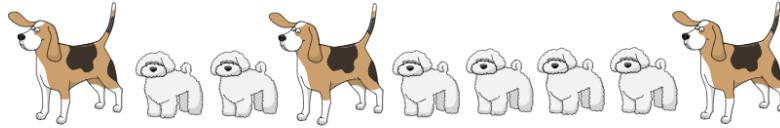
memungkinkan untuk menjalankan satu operasi saja pada satu saat. Oleh karena itu, sistem operasi mempunyai suatu modul perangkat lunak rumit (disebut penjadwal/*scheduler*) yang memutuskan operasi mana yang harus dijalankan berikutnya, dengan mempertimbangkan bahwa keterbatasan sumber daya perangkat keras yang tersedia dan kebergantungan sebuah tugas dengan tugas-tugas lainnya.

https://en.wikipedia.org/wiki/Critical_path_method

https://en.wikipedia.org/wiki/Scheduling_%28computing%29



Terdapat dua jenis anjing yang berbaris seperti tampak dalam gambar berikut ini:



Suatu pertukaran tempat dapat terjadi antara dua anjing yang berdiri bersebelahan. Akan dilakukan beberapa kali pertukaran tempat, sehingga tiga anjing besar akan berdiri dalam tiga posisi yang bersebelahan.

Tantangan:

Berapakah banyaknya pertukaran sesedikit mungkin yang diperlukan?

Pilihan Jawaban:

- 7
- 5
- 6
- 8

Jawaban:

Jawaban yang benar adalah 6.

Ketiga anjing besar akan berdiri dalam tiga posisi yang bersebelahan dengan cara:

- Pertukaran pertama antara anjing besar pertama ke sebelah kanan sebanyak dua kali, kemudian
- Pertukaran anjing besar terakhir ke sebelah kiri sebanyak empat kali.

Setiap anjing kecil harus dilibatkan dalam pertukaran karena setiap anjing kecil diletakkan di antara dua anjing besar. Pertukaran dua anjing kecil tidak akan ada dampaknya sehingga pertukaran harus dilakukan antara satu anjing kecil dengan satu anjing besar. Karena terdapat enam anjing kecil, berarti harus ada paling sedikit enam pertukaran. Perhatikan bahwa percobaan untuk memindahkan ketiga anjing besar ke sebelah kiri atau ke sebelah kanan akan membutuhkan pertukaran lebih dari enam kali.

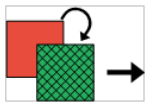
Ini Informatika!

Data tersimpan dalam memori kompute. Hal ini mencakup memori internal yang disebut RAM dan memori eksternal yang misalnya bisa berbentuk hard drive atau USB. Komputer dapat mengakses data dalam memori internal dengan sangat cepat tetapi memori eksternal lebih murah dibandingkan memori interna. Hal ini berarti computer modern biasanya mempunyai lebih banyak memori eksternal, tetapi ilmuwan computer mencoba untuk menemukan cara untuk menggunakan memori internal sedapat mungkin.

Untuk masalah ini, operasi yang dipertimbangkan adalah hanya pertukaran (swap). Jika kita memandang anjing-anjing tersebut sebagai data yang tersimpan di memori computer, maka suatu pertukaran melibatkan perubahan lokasi dari dua buah data. Jika data ini ada di memori eksternal, maka tujuan sesedikit mungkin pertukaran adalah yang kita perlukan untuk melaksanakan seluruh tugas secepat mungkin. Pertukaran juga digunakan dalam beberapa algoritma pengurutan (sorting), yaitu mengurutkan data secara membesar (ascending) atau mengecil (descending). Sebagai contoh dalam bubble sort: dalam setiap pass kita menukar dua data yang bersebelahan yang tidak berurutan.



Milan berhasil mengembangkan robot yang mampu membaca petak berwarna, mengubah warnanya dan berpindah 1 petak ke kiri atau ke kanannya. Robot tersebut bergerak dengan aturan yang digambar dan diartikan sebagai berikut:



Jika dia berada di sebuah petak merah, maka robot mengganti warna petak menjadi hijau dan berpindah 1 petak ke kanan.



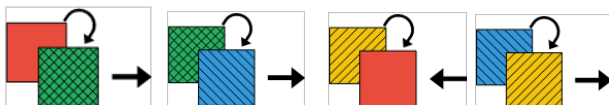
Jika dia berada di sebuah petak merah, maka robot mengganti warna petak menjadi hijau dan berpindah 1 petak ke kiri.

Pada awalnya, robot berada pada petak paling kiri. Ia mendeteksi warna petak, menemukan aturan yang berkaitan dengan warna tersebut, dan berpindah sesuai aturan tersebut. Kemudian robot akan mengulangnya dari petak di mana ia berada. Jika ia tak dapat menemukan aturan, maka berhenti dan keluar dari petak.

Diberikan sederet petak sebagai berikut:



Dan aturannya adalah:



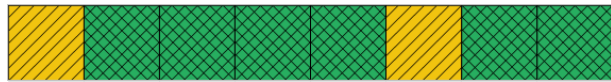
Tantangan:

Pilihlah keadaan petak saat robot berhenti:

- a)
- b)
- c)
- d)

Jawaban:

Jawaban yang benar adalah a)



Ini Informatika!

Dalam bidang informatika, model komputasi memegang peranan sangat penting. Model komputasi adalah aturan dan struktur yang harus ditaati.

Contohnya, model komputasi dari sebuah program komputer (*software*), adalah sekumpulan teks berisi instruksi dalam bahasa programming yang dipakai untuk menulis program tersebut.

Tantangan yang diberikan di atas sangat mirip dengan model mesin Turing. Mesin Turing adalah model yang sangat berguna sebagai model komputasi karena walaupun sederhana, model ini setara dengan berbagai macam bahasa programming. Artinya, kita bisa menerjemahkan berbagai macam program komputer ke dalam mesin Turing, dan sebaliknya.



Copyright 2017 Bebras Indonesia
Licence CC BY-NC-SA 4.0