**Pembahasan TO OSNP LOPI**

**Analitika dan Logika**

1. **Jawaban: 256**

Perhatikan kalimat pertama dan kedua. Agar kalimat pertama TRUE, X[1] dan X[2] keduanya harus TRUE. Sementara itu, agar kalimat kedua TRUE, X[2] harus FALSE dan X[3] harus TRUE. Karena X[2] tidak mungkin TRUE dan FALSE di saat yang bersamaan, maka pasti salah satu dari kalimat pertama ataupun kedua harus FALSE, apapun konfigurasi nilai-nilai X[i]. Maka, semua 28 = 256 konfigurasi memenuhi.

1. **Jawaban: 9** log10(22023) = 608,9837. Berarti, 22023 terdiri atas 608 digit. Untuk mengetahui digit terdepannya, kita dapat mencari-cari nilai log10(*a* ⋅ 10608) dan log10((*a* + 1) ⋅ 10608) yang mengapit nilai log10(22023).

Ingat bahwa log10(*a* ⋅ 10608) = 608 + log10 *a*. Saat mencoba digit 9, didapat bahwa 608 + log10(9) = 608,9542 < 608,9837 < 608 + log10(10) =

609. Oleh karena itu, digit pertama dari 22023 adalah 9.

1. **Jawaban: 10**

Tinjau kasus terburuknya. Sembilan hari pertama, Kwek selalu mengambil kaus kaki merah dan biru sehingga tidak berwarna sama. Hari kesepuluh, tersisa hanya kaus kaki merah, sehingga dapat dijamin bahwa setelah 10 hari, Kwek pernah memakai sepasang kaus kaki berwarna sama.

1. Jawaban: ayam

Perhatikan bahwa ukuran vector s hanya maksimal sebesar 3 × 2023 saja, sehingga program tersebut cukup cepat untuk diketik dan dijalankan. Diperoleh keluaran ayam .

1. **Jawaban: 9**

Pak Dengklek dapat menggunakan *binary search* untuk menentukan bilangan tersebut. Ingat bahwa dengan *binary search*, apabila sekarang terdapat *x* kemungkinan bilangan, setelah satu tebakan akan tersisa ⌊*x*/2⌋ kemungkinan bilangan. Maka, Pak Dengklek harus menebak sebanyak ⌈log2 420⌉ = 9 kali.

1. **Jawaban: 7**

Pak Dengklek dapat menggunakan *binary search* lagi. Perhatikan bahwa indeks dengan nilai maksimum adalah indeks terakhir sehingga nilai elemen itu dikurangkan dengan elemen sebelumnya bernilai positif.

Anggap barisan tersebut sebagai barisan sepanjang 99 dengan A[i] = bilangan (i+1) dikurangi bilangan i. Untuk mendapatkan satu nilai A[i], diperlukan satu tebakan. Maka, dibutuhkan ⌈log2 99⌉ = 7 tebakan.

1. **Jawaban: ABDAN**

Soal ini dapat diselesaikan dengan konsep *nim game*. Sesuai konsep *nim game*, pemain pertama memiliki strategi untuk menang jika dan hanya jika bitwise XOR dari banyaknya batu pada seluruh tumpukan tidak sama dengan 0. Karena XOR dari 21, 11, 20, 6, 13, dan 5 adalah 4, maka Abdan memiliki strategi untuk menang.

1. **Jawaban: 5**

Perhatikan bahwa pernyataan Abdan akan benar apabila hari ini ia jujur dan kemarin ia berbohong, atau sebaliknya. Begitu pula, pernyataan Daniel akan benar apabila hari ini dan kemarin ia keduanya jujur, atau sebailknya. Syaratsyarat tersebut terpenuhi pada hari Senin, Selasa, Rabu, Jumat, dan Sabtu.

Maka, ada 5 kemungkinan.

1. **Jawaban: 4**

Tangki 1 berisi air dari awal.

Tangki 2 akan dialiri air dari tangki 1.

Tangki 3 akan dialiri air dari tangki 1 karena tangki 2 sudah penuh.

Tangki 4 akan dialiri air dari tangki 2 karena tangki 3 sudah penuh.

Tangki 5 dan tangki 6 tidak akan dialiri air, karena mereka saling bergantung satu sama lain.

Maka, ada 4 tangki yang akan penuh setelah proses tersebut selesai.

1. **Jawaban: 8607**

Soal ini juga bisa diselesaikan dengan mencari pola, tetapi ada cara matematisnya.

Misalkan *F*(*x*) = *rx*, maka

*F*(*x*) = 3*F*(*x* − 1) − 2*F*(*x* − 2) *rx* = 3*rx*−1 − 2*rx*−2 *r*2 = 3*r* − 2

*r*2 − 3*r* + 2 = 0

(*r* − 2)(*r* − 1) = 0

sehingga *r* = 2 atau *r* = 1. Maka, *F*(*x*) pasti berbentuk *a* ⋅ 2*x* + *b* ⋅ 1*x* = *a* ⋅ 2*x* + *b*. Masukkan nilai *F*(1) dan *F*(2):

*F*(1) = *a* ⋅ 21 + *b* = 2*a* + *b* = 1

*F*(2) = *a* ⋅ 22 + *b* = 4*a* + *b* = 3

Diperoleh *a* = 1 dan *b* = −1. Maka, *F*(*x*) = 2*x* − 1. Dengan demikian, *F*(202 320 232 023) = 2202 320 232 023 − 1 ≡ 8607 (mod 10000).

**Pemrograman**

# B1. Lelucon Sisipan

**Isian Singkat:**

1. 8
2. 2023
3. 2

**Subsoal 1: U**ntuk semua elemen A, gunakan perulangan untuk mengetahui apakah jumlah dari semua elemen lain sama dengan elemen tersebut. Solusi ini bekerja dalam *O*(*N*2).

**Subsoal 2:** Perhatikan bahwa karena array A hanya berisi bilangan bulat positif, elemen yang disisipkan Kwak pasti merupakan elemen terbesar dari A. Mencari elemen terbesar dapat kita lakukan dengan perulangan biasa dalam *O*(*N*).

# B2. Lelucon Ransel

**Isian Singkat:**

1. 13
2. 11
3. 2047276

**Subsoal 1:** Gunakan brute force berupa bitmask atau rekursi untuk mendapatkan semua nilai kapasitas ransel yang dapat dipenuhkan Pak Dengklek, serupa seperti soal **TO3.B1. Variasi**. Solusi ini berjalan dalam *O*(*Q* ⋅ 2*N*).

**Subsoal 2:** Karena nilai Q[i] hanya sampai dengan 10 000, kita dapat menggunakan DP Knapsack (berisi boolean) untuk mendapatkan nilai kapasitas ransel yang dapat dipenuhkan Pak Dengklek. Misalkan *dp*(*i*,*c*) adalah mungkin atau tidaknya Pak

Dengklek memenuhkan ransel berkapasitas *c* dengan *i* barang pertama. Pada awalnya, *dp*(0,0) = TRUE. Transisinya adalah *dp*(*i*,*c*) = *dp*(*i* − 1,*c* − *Wi*) OR *dp*(*i* − 1,*c*) (mengambil atau tidak mengambil barang ke-*i*).

Hanya ada *O*(*N* ⋅ max*Q*) state yang mungkin, dan transisinya berjalan dalam *O*(1), sehingga kompleksitas waktunya adalah *O*(*N* ⋅ max*Q*). Anda mungkin membutuhkan optimasi *flying table* dalam melakukan DP ini.

# B3. Lelucon Olahraga

Perhatikan bahwa array A akhir yang dibuat Kwak sesungguhnya merupakan array dengan elemen zig-zag (antarelemennya naik, turun, naik, turun, dan seterusnya).

**Isian Singkat:**

1. 3
2. 1
3. 6

**Subsoal 1:** Gunakan brute force berupa bitmask atau rekursi untuk mendapatkan semua kemungkinan array A setelah Kwak menghapus beberapa elemennya, serupa seperti soal **TO3.B1. Variasi**. Setelah itu, cek satu per satu apakah array tersebut zig-zag, dan dapatkan panjang array maksimum yang bisa diperoleh. Solusi ini bekerja dalam *O*(*N* ⋅ 2*N*).

**Subsoal 2:** Gunakan Dynamic Programming. Kita ingin mencari panjang subsequence zig-zag maksimum. Misalkan *A*(*x*) adalah panjang subsequence zigzag maksimum yang elemen terakhirnya *x* dan terakhir menaik, dan *B*(*x*) serupa tetapi terakhir menurun. Transisinya adalah

*A*(*x*) = max⎛1, max (*B*(*y*) + 1)⎞

*y*<*x Ay*<*Ax* ⎠

⎝

*B*(*x*) = max⎛1, max (*A*(*y*) + 1)⎞

*y*<*x*

⎝

*Ay*>*Ax* ⎠

dan jawaban kita adalah *N* − max1≤*i*≤*N*(*A*(*x*),*B*(*x*)). Solusi ini bekerja dalam *O*(*N*2) karena ada *O*(*N*) state dan *O*(*N*) transisi.

# B4. Lelucon Hadiah

**Isian Singkat:**

1. 7
2. 4
3. 19

**Subsoal 1:** Tidak ada solusi khusus untuk subsoal ini. Subsoal ini hanya ada untuk implementasi Subsoal 2 yang kurang cepat untuk mendapatkan poin penuh.

**Subsoal 2:** Perhatikan bahwa pasti garis-garis terluar yang dibuat Pak Dengklek membentuk sebuah poligon konveks. Dengan kata lain, keliling dari poligon itu adalah keliling dari [*convex hull*](https://cp-algorithms.com/geometry/convex-hull.html)semua titik. Gunakan algoritma *convex hull* favorit Anda untuk menyelesaikan soal ini dalam *O*(*N* log *N*).

# B5. Lelucon Terakhir

**Isian Singkat:**

1. 4
2. 17
3. 2

**Subsoal 1:** Perhatikan bahwa kita ingin mencari *X* maksimum sehingga apabila apabila kita hanya memperhitungkan edge-edge yang bobotnya ≥ *X*, *minimum spanning tree* yang terbentuk memiliki total bobot ≤ *S*. Maka, kita dapat mencoba semua kemungkinan *X* (ada *O*(*M*) kemungkinan). Satu kali percobaan memerlukan waktu *O*(*N* log *N*), sehingga kompleksitas waktunya menjadi

*O*(*NM* log *N*).

**Subsoal 2:** Perhatikan observasi bahwa apabila nilai *X* = *x*, dan MST yang terbentuk memiliki total bobot ≤ *S*, maka untuk semua *X* < *x*, MST yang terbentuk juga memiliki total bobot ≤ *S*. Oleh karena itu, kita dapat melakukan *binary search the answer* untuk nilai *X* maksimum yang memenuhi, sehingga kita hanya perlu melakukan *O*(log *M*) kali pengecekan. Kompleksitas waktu akhirnya adalah

*O*(*N* log *N* log *M*).