SCPC Kontes Mingguan 1

16 August 2020

1 Sapi Pemalu

Tag: math, greedy
Pertama anggap tidak ada kandang spesial. Optimalnya untuk meletakkan sapi berjarak tepat K bukan? Jadi jumlah sapi optimalnya ada N/K. Bagaimana jika ada kandang spesial? Pertama-tama semua kandang spesial pasti diisi dengan sapi pada jawaban optimal. Selanjutnya, kita bisa mengisi sapi secara greedy dari kandang terkecil: cari kandang terkecil yang dapat ditempati sapi, kandang tersebut pasti optimal. Ini dapat dilakukan dengan menyimpan kandang terakhir yang berisi sapi lalu terus menerus meletakkan K sapi sampai melewati kandang spesial berikutnya. Jika kandang spesial tersebut berjarak K dari kadang terakhir, letakkan sapi berikutnya pada kandang selanjutnya.

Kompleksitas: O(M)

2 Toni Vs Tere

Tag: If's
Perhatikan makna tersembunyi pada deskripsi.

> "Wahai para pendukungku, apabila kalian percaya kepadaku, maka percayakan kekasihku, Tere, untuk maju sebagai walikota karena aku mencintainya." Bahkan Tere pun mengungkapkan hal yang sama! "Wahai para pendukungku, apabila kalian percaya kepadaku, maka percayakan kekasihku, Toni, untuk maju sebagai walikota karena aku mencintainya."

Jadi jika Toni memiliki X pendukung dan Tere memiliki Y pendukung, maka Toni akan mendapatkan Y suara dan Tere akan mendapatkan X suara. Perhatikan syarat menangnya: "minimal 50% ditambah 1 suara". Jadi, jika misal Toni mendapatkan 10 suara dan Tere mendapatkan 11 suara akan terjadi voting ulang karena 50% + 1 suaranya adalah 11.5. Sehingga, agar tidak terjadi voting ulang, selisih suara Toni dan Tere harus paling tidak 2.

Kompleksitas: O(1)

3 Doom Arena

Tag: BFS
Pertama kita bisa mengurutkan bom secara menaik berdasarkan waktu meledaknya sehingga kita mendapatkan urutan pengunjungan setiap bom. Perhatikan juga karena T_i maksimal 14, jika ada lebih dari 14 bom maka automatis GAME OVER karena tidak mungkin kita mengenai semuanya. Kita bisa melakukan BFS dengan menyimpan state (x, y, bom_ke, time) yang menyatakan posisi dimana, target berikutnya bom ke berapa, dan waktu sekarang. Transisinya, jika saat ini sedang di posisi biasa, kita bisa bergerak dari (x, y, bom_ke, time) ke (x', y', bom_ke, time+1) dimana x' dan y' adalah semua kemungkinan gerakan valid keempat arah. Jika (x, y) adalah posisi bom dan time = $T_{bom\_ke} - 1$, maka kita bergerak ke (x', y' bom_ke+1, time+1) karena *the stalker* akan kena bomnya pada giliran berikutnya. Perhatikan juga jika time ¿ 14, maka kita sudah tidak mungkin menang karena semua bom sudah meledak.

Kompleksitas: O($RCTime^2$)

4 Doom Arena

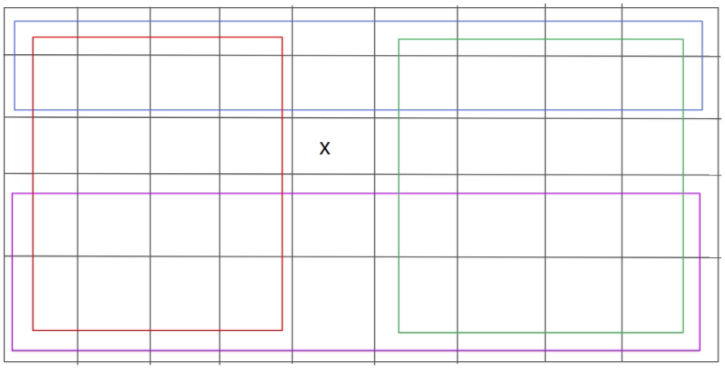
Tag: Greedy
Karena kita ingin meminimalkan selisih elemen terbesar dan terkecil, tentu optimal jika mulai memindahkan karyawan dari gaji terbesar atau terkecil jadi kita sort dulu karyawan berdasarkan gajinya sehingga problem ini berubah menjadi "carilah selisih minimal di semua subarray dengan panjang N - K". Hal ini dapat diselesaikan dengan looping sederhana dengan mencoba-coba semua kemungkinan titik mulai dan titik selesai.

Kompleksitas: O(N)

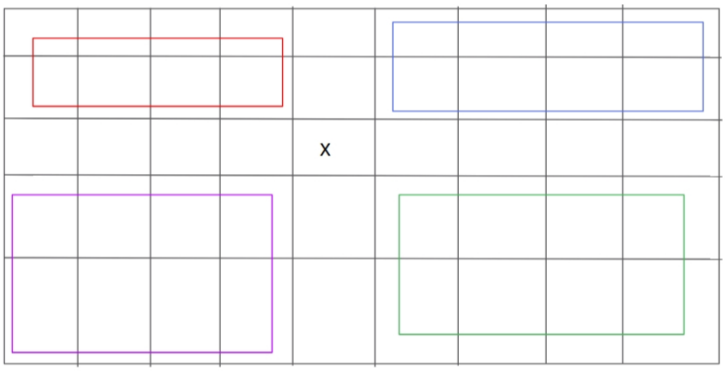
5 Penyakit Tikus Gila

Tag: Math

Untuk setiap petak (i, j), kita bisa menghitung kontribusinya kepada jawaban akhir. Kontribusi suatu petak (i, j) adalah berapa banyak persegi panjang yang dibuat yang menggandung petak ini. Hal ini dapat dihitung sebagai berikut:



Untuk keempat daerah (utara, selatan, timur, barat), memilih 4 titik sembarang akan menghasilkan persegi panjang yang tidak mencangkup titik (i, j). Banyaknya cara melakukan ini adalah C(X, 4) dimana X adalah jumlah titik pada daerah tersebut. Namun, kita melakukan double counting karena ada 4 daerah yang dihitung dua kali, yakni:



Jadi, kita mengurangi jawaban dengan banyaknya cara memilih persegi pada 4 daerah ini. Untuk menghitung banyaknya titik pada suatu persegi panjang (x1, y1, x2, y2), kamu dapat menggunakan prefix sum dua dimensi yang precompute dalam $O(N^2)$ dan menghitung dalam $O(1)$. Jadi setelah menemukan banyaknya persegi yang tidak mengandung petak (i, j), banyaknya persegi yang mengandung petak (i, j) adalah $C(N, 4) - tidak_m engandung$. Karena soal meminta *expected value*, setelah menghitung kontribusi semua petak, jawaban ini kamu bagi dengan $C(N, 4)$, banyaknya total cara memilih persegi.

Kompleksitas: O(1) karena ukuran petaknya fixed 1000 x 1000

6 Queue Pole

Tag: Tree, DFS
Kita modelkan ini sebagai permasalahhan graph, dengan pole sebagi vertex dan tali sebagai edge. Pertama-tama, di cek dulu apakah semua vertex memiliki degree ¡= 4. Jika iya, kita bisa melihat bahwa graph dapat dibuat menggunakan queue pole hanya dua jenis, yaitu:

1. graph berbentuk tree
2. graph berbentuk pseudotree, yaitu graph yang hanya memiliki 1 cycle.

Kamu bisa mengecek kedua hal ini sekaligus dengan cara: DFS dari sembarang titik, anggap sekarang berada di vertex u. Untuk semua vertex v yang dapat dikunjungi u, di lakukan dua hal berikut:

1. jika v parent dari u, abaikan.
2. Jika v sudah dikunjungi sebelumnya, kita mendapatkan cycle
3. jika v belum dikunjungi sebelumnya, kita mengunjungi v.

Jika poin pertama terjadi 0 kali, maka graph tree, jika poin pertama terjadi sekali, maka graph pseudotree. Perhatikan bahwa mungkin ada lebih dari satu connected component jadi di lakukan algoritma ini untuk setiap connected component.

Kompleksitas: O(N + M)

7 Panggilan Akrab

Tag: Brute Force
Perhatikan bahwa banyaknya kemungkinan panggilan akrab paling banyak 120N. Sehingga untuk setiap orang, kita bisa mengecek secara brute force untuk setiap kemungkinan string apakah sebelumnya sudah ada orang yang memiliki kemungkinan string ini. Kamu bisa menyimpan panggilan-panggilan yang sudah ditemukan pada sebuah data structure yang memudahkan pencarian, misalkan *hashmap* atau *treemap*.

Kompleksitas: O(NS), dimana S adalah panjang input

8 Berti Si Galau

Tag: Combinatorics
Kita bisa mengubah X menjadi representasi binernya lalu menyelesaikan setiap bit secara independen. Anggap bit ke-i nya 1, jadi kita harus mencari banyaknya cara set 0/1 pada N digit binary sehingga xornya = 1, ini sama saja dengan banyaknya cara set 0/1 pada N dimana jumlah 1nya ganjil. Karena N hanya 100.000, kamu bisa menghitung ini dengan kombinasi:

$$s_1 = C(N, 1) + C(N, 3) + C(N, 5) + ...$$

Logika sama jika bit ke-inya 0 yang sama saja dengan mencari banyaknya cara set sehingga jumlah 1nya genap.

$$s_0 = C(N, 0) + C(N, 2) + C(N, 4) + ...$$

Perhatikan bahwa kita hanya perlu menghitung S sekali dan nilainya dapat dipakai untuk menghitung semua bit dari jawaban. Hasil akhirnya adalah perkalian dari semua bit, jika bit ke-inya 0, kalikan jawaban dengan s_0, jika bit ke-inya 1, kalikan jawaban dengan s_1.

Untuk menghitung kombinasi dengan cepat, kamu bisa memprecompute array fact[i] dimana fact[i] adalah i! mod 1.000.000.007, dan inv[i], dimana inv[i] adalah 1/i! mod 1.000.000.007. C(N, K) = fact[n]*inv[k]*inv[n-k].

Kompleksitas: O(N)

9 Saluran Televisi

Tag: Ad Hoc

1. Jika operasi 'next' dan saluran berikutnya X, maka saluran sebelumnya X-1.
2. Jika operasi 'prev' dan saluran berikutnya X, maka saluran sebelumnya X+1.

Perhatikan kemungkinan siklik. Jika X - 1 = 0, maka jawabannya 99, jika X + 1 = 100, maka jawabannya 0.

Kompleksitas: O(1)