

Software Engineering Competition (SECOMP)

Penyisihan

D3 Rekayasa Perangkat Lunak Aplikasi
Telkom University

26 Juli 2025

Peraturan

- Setiap peserta akan mengerjakan soal sesuai dengan waktu yang sudah ditentukan.
- Dilarang untuk membuka internet, atau referensi apapun selama lomba berlangsung.
- Dilarang untuk mencontek peserta lain.
- Segala bentuk penggunaan struktur data dan library built-in diperbolehkan, namun dilarang menggunakan library eksternal.

Foreword

Selamat datang di SECOMP. Apakah kalian gugup? Tenang saja, karena gugup adalah suatu respon dari otak terhadap ancaman yang ada di depan. Apakah kalian mengira kalau soal babak penyisihan ini adalah ancaman bagi kalian? Mungkin saja begitu. Tapi tenang, karena perlombaan ini hanyalah sebuah peristiwa kecil dalam hidup kalian yang masih panjang. Karena itu, tenanglah. Apapun hasilnya, yang penting adalah apa yang kalian lakukan kedepannya. Tidak puas dengan hasilnya? Berlatih lagi sampai puas. Karena itu, tenang dan tarik nafas. Kalian akan masuk ke dalam dunia Competitive Programming yang membutuhkan kejernihan pikiran kalian.

1 Kerakusan Manusia

"Aku menginginkan ini; setelahnya, aku tidak menginginkan apa-apa lagi..."

Di dalam lubuk hati setiap manusia, selalu ada hasrat yang meronta untuk lebih. Setelah satu keinginan terpenuhi, muncul lagi keinginan berikutnya, seakan tak pernah ada penghentian. Kerakusan manusia tak ubahnya seperti sebuah tambang yang tak pernah padam: setiap kali kita menggali, kita mendapatkan kepuasan, namun cadangan kepuasan itu perlahan menipis.

Setiap tambang dalam problem ini merepresentasikan satu aspek kebutuhan atau keinginan—misal kekayaan, ketenaran, atau kenikmatan. Nilai melambangkan kadar kepuasan maksimal yang dapat digali pada satu pengambilan pertama, sementara menunjukkan seberapa cepat kepuasan itu memudar setelah setiap penggalian. Dengan demikian, tambang yang kaya akan kepuasan tetapi cepat habis menggambarkan nafsu yang besar namun mudah jenuh, sedangkan tambang yang lambat penurunan daya pikatnya mencerminkan keinginan yang terus memanggil meski telah terpenuhi.

1.1 Format Input

- Baris pertama berisi dua bilangan bulat N dan K : jumlah tambang dan jumlah pengambilan total yang boleh dilakukan.
- Baris berikutnya masing-masing berisi dua bilangan bulat A_i dan D_i untuk $i = 1, 2, \dots, N$:
 - A_i : kepuasan awal dari tambang ke- i saat satuan pertama diambil.
 - D_i : penurunan kepuasan setiap kali satuan diambil dari tambang ke- i .

1.2 Format Output

Satu bilangan bulat: jumlah kepuasan maksimum yang dapat dikumpulkan setelah melakukan tepat K pengambilan.

1.3 Contoh Input

Input 1

3 5

5 1

3 1

4 2

Input 2

3 7

20 8

9 2

6 1

1.4 Contoh Output

Output 1

19

Output 2

64

1.5 Batasan dan Catatan

- $1 \leq N \leq 2 \times 10^5$
- $1 \leq K \leq 2 \times 10^5$
- $0 \leq A_i \leq 10^9$
- $1 \leq D_i \leq 10^9$
- Jika pada suatu tambang nilai kepuasan turun menjadi nol, pengambilan selanjutnya dari tambang tersebut tidak memberikan kepuasan lagi.
- Diharapkan solusi menggunakan algoritma **greedy** dengan struktur data *max-heap* (priority queue) untuk efisiensi.

1.6 Penjelasan Contoh 1

Misalkan ada tiga tambang:

- Tambang 1: $A_1 = 5$, $D_1 = 1$
- Tambang 2: $A_2 = 3$, $D_2 = 1$
- Tambang 3: $A_3 = 4$, $D_3 = 2$

Kita boleh mengambil $K = 5$ kali. Setiap kali, kita memilih tambang dengan *kepuasan saat ini* tertinggi:

1. Ambil dari tambang 1: dapat 5 (sisa 4)
2. Ambil dari tambang 3: dapat 4 (sisa 2)
3. Ambil dari tambang 1: dapat 4 (sisa 3)
4. Ambil dari tambang 1: dapat 3 (sisa 2)
5. Ambil dari tambang 2: dapat 3 (sisa 2)

Total kepuasan = $5 + 4 + 4 + 3 + 3 = 19$.

1.7 Penjelasan Contoh 2

Misalkan ada tiga tambang:

- Tambang 1: $A_1 = 20$, $D_1 = 8$
- Tambang 2: $A_2 = 9$, $D_2 = 2$
- Tambang 3: $A_3 = 6$, $D_3 = 1$

Kita boleh mengambil $K = 7$ kali. Setiap kali, kita memilih tambang dengan *kepuasan saat ini* tertinggi:

1. Ambil dari tambang 1: dapat 20, sisa kepuasan 12

2. Ambil dari tambang 1: dapat 12, sisa kepuasan 4
3. Ambil dari tambang 2: dapat 9, sisa kepuasan 7
4. Ambil dari tambang 2: dapat 7, sisa kepuasan 5
5. Ambil dari tambang 3: dapat 6, sisa kepuasan 5
6. Ambil dari tambang 2: dapat 5, sisa kepuasan 3
7. Ambil dari tambang 3: dapat 5, sisa kepuasan 4

Total kepuasan = $20 + 12 + 9 + 7 + 6 + 5 + 5 = 64$.

2 Apakah "itu" ada?

"Sesuatu", yang bisa digunakan untuk menyebutkan apapun, terutama saat kita tidak mengetahui atau tidak ingin mendefinisikan "sesuatu" tersebut. Coba berpikir, apakah "sesuatu" itu benar-benar ada sebelumnya, atau ada hanya ketika kita mengamatinya? Sebagai contoh, bila kamu menonton sebuah film, apakah film tersebut memang memiliki cerita seperti itu, bahkan sebelum kamu menontonnya? Atau cerita film tersebut sebenarnya berubah saat kamu menontonnya? Apakah dunia ini beradaptasi kepadamu?

Di sini, terdapat sebuah konsep tentang "sesuatu" yang hanya ada ketika diamati. Bayangkan sebuah sistem di mana objek-objek dalam suatu ruang hanya "muncul" atau "terwujud" ketika ada pengamat yang melihatnya. Kamu diberikan sebuah grid 2D berukuran $N \times M$ yang berisi berbagai objek. Setiap sel dalam grid dapat berisi:

- . (kosong)
- # (objek yang selalu ada)
- ? (objek yang hanya ada ketika diamati)

Terdapat K pengamat yang masing-masing berada di posisi tertentu dan dapat melihat dalam 4 arah (atas, bawah, kiri, kanan) dengan jarak pandang tak terbatas hingga terhalang oleh # atau batas grid. Kamu harus menentukan berapa banyak objek ? yang "ada" (terwujud) karena diamati oleh minimal satu pengamat.

Format Input

1. Baris pertama: tiga integer N, M, K ($1 \leq N, M \leq 1000, 1 \leq K \leq 100$)
2. N baris berikutnya: string sepanjang M karakter yang merepresentasikan grid
3. K baris berikutnya: dua integer x_i, y_i ($1 \leq x_i \leq N, 1 \leq y_i \leq M$), posisi pengamat ke- i

Format Output

Cetak satu bilangan bulat: jumlah objek ? yang diamati oleh minimal satu pengamat.

Contoh Input

Input 1

3 4 2

?..?

....

?..?

1 1

2 2

Input 2

..#?

....

...?

1 2

2 4

Contoh Output

Output 1

3

Output 2

2

Batasan dan Catatan

- Pengamat dapat berada di sel ., ?, atau #
- Objek # menghalangi pandangan, sedangkan ? dan . tidak menghalangi
- Pengamat dapat melihat objek ? di selnya sendiri jika posisinya adalah ?
- Setiap ? yang diamati dihitung sekali saja, meski diamati oleh beberapa pengamat
- Pandangan pengamat berhenti ketika mencapai # atau batas grid

Penjelasan Contoh 1

Grid dengan koordinat (baris, kolom) dimulai dari (1,1):

```
  1 2 3 4
1 ? . . ?
2 . . . .
3 ? . . ?
```

Analisis pengamat:

- **Pengamat di (1,1):** Berada di sel ?
 - Posisi sendiri: ? di (1,1) → terwujud

- Ke atas: terhalang batas grid
- Ke bawah: melihat (2,1), (3,1) yang berisi ., ? \rightarrow ? di (3,1) terwujud
- Ke kiri: terhalang batas grid
- Ke kanan: melihat (1,2), (1,3), (1,4) yang berisi ., ., ? \rightarrow ? di (1,4) terwujud

• **Pengamat di (2,2):** Berada di sel .

- Ke atas: melihat (1,2) yang berisi . \rightarrow tidak ada ?
- Ke bawah: melihat (3,2) yang berisi . \rightarrow tidak ada ?
- Ke kiri: melihat (2,1) yang berisi . \rightarrow tidak ada ?
- Ke kanan: melihat (2,3), (2,4) yang berisi ., . \rightarrow tidak ada ?

Objek ? yang terlihat:

- ? di (1,1) \rightarrow terlihat oleh pengamat di (1,1)
- ? di (1,4) \rightarrow terlihat oleh pengamat di (1,1)
- ? di (3,1) \rightarrow terlihat oleh pengamat di (1,1)
- ? di (3,4) \rightarrow tidak terlihat oleh pengamat manapun

Total objek ? yang terwujud: **3**

Penjelasan Contoh 2

Grid dengan koordinat (baris, kolom) dimulai dari (1,1):

	1	2	3	4
1	.	.	#	?
2
3	.	.	.	?

Analisis pengamat:

• **Pengamat di (1,2):** Berada di sel .

- Ke atas: terhalang batas grid.
- Ke bawah: melihat (2,2), (3,2), (4,2) yang berisi . seluruhnya \rightarrow tidak ada ?.
- Ke kiri: (1,1) yang berisi . \rightarrow tidak ada ?.
- Ke kanan: melihat (1,3), (1,4) yang berisi #, ? \rightarrow terhalang oleh #.

• **Pengamat di (2,4):** Berada di sel .

- Ke atas: melihat (1,4) yang berisi ? \rightarrow ? terwujud.
- Ke bawah: melihat (3,4) yang berisi ? \rightarrow ? terwujud.
- Ke kiri: melihat (2,3), (2,2), (2,1) yang berisi . seluruhnya \rightarrow tidak ada ?
- Ke kanan: terhalang batas grid.

Objek ? yang terlihat:

- ? di $(1,4) \rightarrow$ terlihat oleh pengamat di $(2,4)$
- ? di $(3,4) \rightarrow$ terlihat oleh pengamat di $(2,4)$
- Pengamat di $(1,2)$ tidak melihat ? apapun.

Total objek ? yang terwujud: **2**