

Assalamualaikum wr wb.

Mohammad Hanif Furqan Aufa Putra

5025221161

TUGAS 7 PBO C

Roads and Libraries

⌚ Time limit **1** second

💾 Memory limit **128** MiB

The cities of Azerbaijan are numbered with numbers from 1 to n . Currently, no city has libraries and no two cities are connected. Two-way roads can be built between pairs of cities that are specified in the input data. The cost of building a library in a city is lib . The cost of building one road is $road$.

A citizen has access to the library if:

- There is a library in his city.
- He can follow the roads from his city to a city that has a library.

Determine the minimum cost of providing access to libraries to all citizens of Azerbaijan.

Input data

The first line contains the number of test cases t ($1 \leq t \leq 10$).

The first line of each test case contains four integers that describe the number of cities n ($1 \leq n \leq 105$), number of roads m ($1 \leq m \leq 105$), cost of a library lib ($1 \leq lib \leq 105$) and cost of a road $road$ ($1 \leq road \leq 105$). Each of the next m lines contains two integers u and v , that describe a bidirectional road that can be built to connect cities u and v . Each road connects two distinct cities.

Output data

Print the minimal cost to provide library access to all citizens of Azerbaijan.

Examples

Input example #1

```
2
3 3 2 1
1 2
3 1
2 3
6 6 2 5
1 3
3 4
2 4
1 2
```

2 3
5 6

Output example #1

4
12

Masalah dalam kode yang diberikan adalah menentukan biaya minimum untuk memberikan akses ke perpustakaan bagi semua warga Azerbaijan. Di sini, terdapat beberapa kota di Azerbaijan yang diwakili oleh angka dari 1 hingga n . Awalnya, tidak ada perpustakaan di kota mana pun, dan tidak ada dua kota yang terhubung. Kita bisa membangun jalan dua arah antara dua kota dengan biaya tertentu. Selain itu, kita juga bisa membangun perpustakaan di kota dengan biaya tertentu. Setiap warga dapat mengakses perpustakaan jika ada perpustakaan di kota mereka atau jika mereka dapat mengikuti jalan-jalan dari kota mereka ke kota lain yang memiliki perpustakaan.

Cara menyelesaikan masalah ini adalah dengan menggunakan struktur data Union-Find (Disjoint Set Union - DSU) untuk mengelompokkan kota-kota yang terhubung menjadi satu kesatuan. Kemudian, kita akan mencari biaya minimum untuk memberikan akses ke perpustakaan dengan cara berikut:

- 1) Inisialisasi semua kota sebagai kelompok terpisah.
- 2) Menghubungkan kota-kota yang terhubung dengan membangun jalan antara kota-kota yang terhubung. Ini dilakukan dengan bantuan DSU (Union-Find).
- 3) Menghitung jumlah kelompok terhubung (ccs) yang terbentuk setelah langkah kedua.
- 4) Selanjutnya, kita akan mencari biaya minimum yang dibutuhkan untuk memberikan akses ke perpustakaan. Ada dua pilihan: membangun perpustakaan di setiap kelompok atau membangun jalan untuk menghubungkan kelompok-kelompok ini dengan satu perpustakaan. Kita akan membandingkan biaya dari kedua pilihan ini dan memilih yang lebih murah.

Struktur data Union-Find (DSU) digunakan untuk mengelompokkan kota-kota yang terhubung. Ini memungkinkan kita untuk dengan cepat menemukan apakah dua kota berada dalam kelompok yang sama atau tidak, serta menggabungkan dua kelompok jika kita membangun jalan yang menghubungkannya.

Perbedaan antara struct dan class dalam kode terletak pada aksesibilitas data. Dalam struct, semua data dideklarasikan sebagai publik secara default, sehingga dapat diakses langsung dari luar struct. Sementara dalam class, data defaultnya adalah pribadi, dan kita perlu menggunakan metode atau fungsi untuk mengakses atau memodifikasi data tersebut. Struct dalam kode didefinisikan untuk UnionFind tanpa memiliki metode atau fungsi tambahan.

DSU adalah singkatan dari "Disjoint Set Union," yang digunakan dalam kode untuk mengelompokkan kota-kota yang terhubung menjadi kelompok yang sama. Ini membantu dalam mencari tahu apakah dua kota berada dalam kelompok yang sama atau tidak. Kruskal adalah algoritma yang bisa digunakan untuk menentukan rangkaian minimal (minimum spanning tree) dalam graf yang menghubungkan semua kota. Dalam kode, DSU digunakan untuk mengelompokkan kota-kota, sementara Kruskal tidak digunakan.

Kode bekerja dengan menghubungkan kota-kota yang terhubung dengan DSU. Kemudian, ia menghitung jumlah kelompok terhubung (ccs) yang terbentuk setelah itu. Selanjutnya, kode mencari biaya minimum untuk memberikan akses ke perpustakaan dengan membandingkan biaya membangun perpustakaan di setiap kelompok dengan biaya membangun jalan untuk menghubungkan kelompok-kelompok tersebut dengan satu perpustakaan. Kode kemudian mencetak biaya minimum yang ditemukan.

Dari kedua kode yang diberikan, meskipun ada perbedaan dalam cara implementasi struktur data Union-Find (DSU), keduanya menghasilkan output yang sama dan memiliki kompleksitas waktu yang sama untuk menyelesaikan masalah. Perbedaan yang ada hanya berkaitan dengan penggunaan struktur data DSU, yang dalam kasus pertama diimplementasikan sebagai sebuah struct, sedangkan dalam kasus kedua diimplementasikan sebagai sebuah class.

Namun, perbedaan dalam penggunaan memori adalah hal yang wajar dan mungkin tidak signifikan. Setiap kompiler atau lingkungan pemrograman memiliki perbedaan dalam alokasi memori dan pengelolaan memori yang digunakan oleh struktur data. Jadi, perbedaan antara 14.7 MiB dan 15 MiB mungkin disebabkan oleh perbedaan dalam pengelolaan memori oleh kompiler atau lingkungan tempat kode dijalankan.

Kedua kode ini secara esensial memiliki algoritma yang sama dan memberikan hasil yang identik, dengan perbedaan hanya pada bagian implementasi struktur data DSU. Oleh karena itu, perbedaan dalam penggunaan memori yang tidak signifikan adalah hal yang wajar dan tidak mempengaruhi hasil akhir yang diberikan oleh kedua kode tersebut.

14979924	Nov 3, 2023, 10:32:47 AM	C++ 11 (gnu 10.2)	15 ms	15 ms	472	✓ Accepted
14979921	Nov 3, 2023, 10:32:27 AM	C++ 11 (gnu 10.2)	15 ms	15 ms	476	✓ Accepted
14979918	Nov 3, 2023, 10:32:15 AM	C++ 11 (gnu 10.2)	15 ms	14 ms	472	✓ Accepted
14979916	Nov 3, 2023, 10:32:00 AM	C++ 11 (gnu 10.2)	15 ms	14 ms	476	✓ Accepted
14979913	Nov 3, 2023, 10:31:47 AM	C++ 11 (gnu 10.2)	15 ms	14 ms	476	✓ Accepted
14979882	Nov 3, 2023, 10:27:03 AM	C++ 17 (gnu 10.2)	15 ms	15 ms	476	✓ Accepted
14979881	Nov 3, 2023, 10:26:49 AM	C++ 17 (gnu 10.2)	15 ms	15 ms	476	✓ Accepted
14979880	Nov 3, 2023, 10:26:41 AM	C++ 17 (gnu 10.2)	15 ms	15 ms	476	✓ Accepted
14979879	Nov 3, 2023, 10:26:32 AM	C++ 17 (gnu 10.2)	15 ms	15 ms	476	✓ Accepted
14979877	Nov 3, 2023, 10:25:55 AM	C++ 17 (gnu 10.2)	15 ms	15 ms	472	✓ Accepted

Struct Code

14987339	Nov 3, 2023, 10:29:19 PM	C++ 17 (gnu 10.2)	15 ms	15 ms	476	✓ Accepted
14987337	Nov 3, 2023, 10:29:12 PM	C++ 17 (gnu 10.2)	15 ms	15 ms	476	✓ Accepted
14987335	Nov 3, 2023, 10:29:06 PM	C++ 17 (gnu 10.2)	15 ms	15 ms	476	✓ Accepted
14987333	Nov 3, 2023, 10:29:00 PM	C++ 17 (gnu 10.2)	15 ms	15 ms	476	✓ Accepted
14987331	Nov 3, 2023, 10:28:55 PM	C++ 17 (gnu 10.2)	15 ms	15 ms	476	✓ Accepted
14987328	Nov 3, 2023, 10:28:48 PM	C++ 17 (gnu 10.2)	15 ms	15 ms	476	✓ Accepted
14987327	Nov 3, 2023, 10:28:43 PM	C++ 17 (gnu 10.2)	15 ms	15 ms	476	✓ Accepted
14987326	Nov 3, 2023, 10:28:37 PM	C++ 17 (gnu 10.2)	15 ms	15 ms	476	✓ Accepted
14987325	Nov 3, 2023, 10:28:32 PM	C++ 17 (gnu 10.2)	15 ms	15 ms	472	✓ Accepted
14987127	Nov 3, 2023, 10:09:42 PM	C++ 17 (gnu 10.2)	15 ms	15 ms	476	✓ Accepted

Class Code

Terima Kasih telah membaca, Waalaikumsalam wr wb.