

Tugas 7 PBO C

Brendan Timothy Mannuel

5025221177

Roads and Libraries

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 ($1 \leq n \leq 10^5$), number of roads m ($1 \leq m \leq 10^5$), cost of a library lib ($1 \leq lib \leq 10^5$) and cost of a road $road$ ($1 \leq road \leq 10^5$). 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.

Pada soal ini kita diperlukan untuk mencari minimum cost dari suatu kota dimana penduduk di kota tersebut harus dapat mempunyai akses ke perpustakaan, diberikan dua cara yaitu dengan membangun perpustakaan di kota tersebut atau dengan membangun jalan untuk menghubungkan 2 kota, kita akan diberikan cost jika kita akan membangun jalan ataupun membangun perpustakaan, kita akan diberikan list untuk jalan mana saja yang dapat kita buat.

Kita dapat menyelesaikan soal ini dengan menggunakan implementasi dari UFDS atau Union-Find Disjoint Sets, algo ini berguna untuk mencari tahu suatu banyaknya segmentasi atau kluster dari suatu graph sehingga kita dapat mencari tahu berapa banyak perpustakaan yang minimal kita harus buat, Serta algo ini juga membantu kita dalam membuat MST atau Minimum Spanning Tree yang berguna untuk menentukan jalan mana saja yang harus dibuat dan mana yang tidak perlu.

```
9 struct UnionFind {
10     vector<int> data;
11     void init (int n){data.assign(n, -1); }
12     bool unionSet(int x, int y){
13         x = root(x); y = root(y);
14         if (x != y){
15             if (data[y] < data[x]) swap(x,y);
16             data[x] += data[y]; data[y] = x;
17         }
18         return x != y;
19     }
20     bool findset(int x, int y) {return root(x) == root(y); }
21     int root(int x){ return data[x] < 0 ? x : data[x] = root(data[x]); }
22     int size(int x){ return -data[root(x)]; }
23 };
```

Disini kita akan mengimplementasikan Struct, kita akan menyimpan data dalam bentuk vector agar data bersifat dinamis, terdapat juga berbagai fungsi yaitu :

1. Init : Fungsi ini digunakan untuk memasukkan data kedalam struct dan membuat root memiliki nilai -1 yang berarti root menuju ke dirinya sendiri.
2. unionSet : Fungsi ini digunakan untuk menghubungkan 2 node yang memiliki parent yang berbeda sehingga dapat menyatu sebagai suatu graph dalam satu segmentasi atau kluster
3. Findset : Fungsi ini digunakan untuk mencari apakah 2 nilai node yang dicari termasuk dalam suatu segmentasi atau kluster yang sama
4. Root : Fungsi ini berguna untuk mengembalikan nilai root, jika dibawah 0 atau yang berarti -1 maka node tersebut merupakan parent dari dirinya sendiri, jika tidak maka kita akan melakukan operasi secara rekursif untuk mencari parent atau root node dari node tersebut.
5. Size : Fungsi ini digunakan untuk mereturn ukuran dari suatu graph yang terhubung.

Berdasarkan struct diatas maka kita dapat menyelesaikan soal berikut seperti berikut

```
35 int main(){
36     int T;
37     Get(T);
38     for (int ii = 0; ii < T; ++ii){
39         int n; int m; int clib; int croad;
40         Get(n); Get(m); Get(clib); Get(croad);
41         UnionFind DSU; DSU.init(n);
42         rep (i, m){
43             int u; int v;
44             Get(u); Get(v);
45             --u; --v;
46             DSU.unionSet(u, v);
47         }
48         int ccs = 0;
49         rep (i, n)
50             ccs += DSU.root(i) == i;
51         ll ans = INFL;
52         for (int i = ccs; i <= n; ++i)
53             MIN(ans, (ll)clib * i + (ll)croad * (n - i));
54         printf("%lld\n", ans);
55     }
56     return 0;
57 }
```

Tentunya yang pertama kita akan menginput berapa banyak testcase yang akan dilakukan, kemudian kita akan melakukan perulangan hingga loop selesai. Kita akan menginputkan berapa banyak kota, berapa banyak jalan, cost dari Pembangunan jalan, dan cost dari Pembangunan perpustakaan. Kemudian kita akan melakukan init sebanyak kota yang ada. Kemudian kita akan melakukan unionset untuk setiap jalan yang diberikan oleh soal. Kemudian kita akan melakukan pengecekan berapa banyak segmentasi atau kluster yang terbentuk dan jumlah akan disimpan dalam CCS atau Connected Components. Kemudian kita akan mulai menghitung jawaban minimal dengan menggunakan rumus :

$$\text{Banyak Perpustakaan} * \text{Cost Perpustakaan} + (\text{Banyak Kota} - \text{Banyak Perpustakaan}) * \text{Cost Perpustakaan}$$

Kenapa I dapat diartikan sebagai banyak perpustakaan karena Jumlah dari CCS atau banyaknya segmentasi menunjukan jumlah minimal perpustakaan yang dibentuk karena didalam satu segmentasi atau kluster diperlukan 1 perpustakaan agar semua kota dapat memiliki akses ke perpustakaan.

Contoh Pengaplikasian:

Pada contoh input, testcase ke 2:

Terdapat 6 kota, 6 jalan, cost membuat perpustakaan = 2, cost membuat jalan = 6

Setelah melakukan input jalan, terdapat 2 kluster yaitu :

Kluster 1 : Kota 1, 2, 3, 4

Kluster 2 : Kota 5 dan 6

Sehingga minimal kita akan membangun 2 Perpustakaan, kemudian kita akan memasukan rumus:

Banyak Perpustakaan * Cost Perpustakaan + (Banyak Kota – Banyak Perpustakaan) * Cost Perpustakaan

Setelah mencoba terlihat bahwa ternyata cost paling minimal dicapai Ketika kita membangun perpustakaan disetiapa kota dikarenakan harga dari membangun perpustakaan jauh lebih murah dibandingkan membangun sebuah jalan.

Kita juga dapat menggunakan Class dalam mendeklarasikan fungsi seperti dibawah ini

```
9 class UnionFind {
10     vector<int> data;
11     public :
12     void init (int n){data.assign(n, -1); }
13     bool unionSet(int x, int y){
14         x = root(x); y = root(y);
15         if (x != y){
16             if (data[y] < data[x]) swap(x,y);
17             data[x] += data[y]; data[y] = x;
18         }
19         return x != y;
20     }
21     bool findset(int x, int y) {return root(x) == root(y); }
22     int root(int x){ return data[x] < 0 ? x : data[x] = root(data[x]); }
23     int size(int x){ return -data[root(x)]; }
24 };
```

Perbedaan yang dapat dihighlight adalah Ketika kita menggunakan struct maka fungsi yang terdapat dapat diakses secara public, sedangkan untuk class fungsi hanya bisa diakses secara private sedangkan jika ingin dibuat public maka kita harus memilih fungsi mana saja seperti diatas.

