

[Home](#) > [B](#) > [A. Parade Panjang](#)

solved

Next >

A 2 / 2

B 3 / 3

C 1 / 3

D 0 / 3

Time limit: 1 s • Memory limit: 64 MB

Indonesian (id)

Deskripsi

Ada parade di Kota Lumino!

Kota Lumino terdiri dari N distrik dan semua pasang distrik memiliki satu jalan yang menghubunginya. Tetapi karena ada parade, Walikota Kota Lumino memutuskan untuk menutup beberapa jalan untuk alasan keamanan. Ia juga memutuskan untuk membuat semua jalan menjadi satu arah sedemikian hingga dalam setiap rute, tidak dimungkinkan untuk kembali ke distrik yang sebelumnya sudah dikunjungi, agar dapat menghindari kemacetan. Dengan kata lain, Kota Lumino sekarang memiliki M jalan satu arah, dimana jalan ke- i berasal dari distrik A_i dan berakhir di distrik B_i , serta memiliki waktu tempuh T_i .

Parade akan dimulai dari distrik 1 dan berakhir pada distrik N . Anda sebagai perencana parade diberikan detail rencana penutupan jalan Kota Lumino. Anda diminta untuk merencanakan rute yang akan ditempuh agar parade berjalan selama mungkin, lalu melaporkan waktu yang diperlukan parade untuk sampai ke distrik N kepada walikota Kota Lumino.

Format Masukan

Baris pertama berisi dua bilangan bulat N dan M , jumlah distrik dan jalan satu arah yang dibuka di Kota Lumino.

M Baris berikutnya masing-masing berisi tiga bilangan bulat A_i , B_i , dan T_i , yang menyatakan bahwa jalan ke- i berasal dari distrik A_i dan berakhir di distrik B_i dengan waktu tempuh T_i .

Format Keluaran

Keluarkan sebuah baris berisi satu bilangan bulat X , waktu terlama yang diperlukan untuk sampai ke distrik N . Jika tidak terdapat cara, keluarkan -1 .

Contoh Masukan 1

```
3 2
1 2 10
2 3 7
```

Contoh Keluaran 1

```
17
```

Code

Submissions

C++20

```
61         int district = neighbour.t1,
62         int weight = neighbour.ss;
63
64         int path_neighbour = dfsRec(adj, memo, district, N);
65
66         if(path_neighbour != -1e9){
67             max_path = max(max_path, weight + path_neighbour);
68         }
69     }
70
71     memo[s] = max_path;
72     return max_path;
73 }
74
75 int DFS(vector<vector<pii>> &adj){
76     vi memo(adj.size(), -1);
77     int last_node = adj.size()-1;
78
79     int result = dfsRec(adj, memo, 1, last_node);
80
81     if(result == -1e9) return -1;
82     return result;
83 }
84 }
85
86 int32_t main(){
87     //c_p_c();
88     ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
89
90     int N, M; cin >> N >> M;
91
92     vector<vector<pii>> adj(N+1);
93     REP(i, M){
94         int u, v, w; cin >> u >> v >> w;
95         adj[u].push_back({v, w});
96     }
97
98     int jalan = DFS(adj);
99
100     cout << jalan;
101
102     return 0;
103 };
```

Contoh Masukan 2

```
3 3
1 2 10
2 3 7
1 3 18
```

Contoh Keluaran 2

```
18
```

Contoh Masukan 3

```
4 3
1 2 5
1 3 7
2 3 4
```

Contoh Keluaran 3

```
-1
```

Penjelasan

Pada contoh pertama, hanya terdapat satu rute menuju distrik 3, yaitu 1->2->3 dengan total waktu sebesar 17.

Pada contoh kedua, rute terlama menuju distrik 3 adalah 1->3 dengan total waktu sebesar 18.

Pada contoh ketiga, tidak ada rute yang dapat digunakan untuk mencapai distrik 4.

Batasan

- $1 \leq N \leq 100.000$
- $1 \leq M \leq \min(100.000, N*(N-1)/2)$
- $1 \leq A_i, B_i \leq N$
- $1 \leq T_i \leq 10^9$
- Dijamin dalam setiap rute tidak dimungkinkan sebuah distrik dapat dikunjungi jika sebelumnya sudah dikunjungi

[Submit](#)[Next >](#)