

See You Soon!

By R. Rafif Aqil Aabid Hermawan

Time Limit : 1s



Description :

It's not a Goodbye it's a See you Soon!

Jadi aku ingin ngesurprise temenku nih sebelum aku sama temenku semua pergi keluar negeri (Pengen Double Degree ternyata 😬). Nah aku sudah punya peta dengan Lokasi mereka sih (dalam bentuk graph tentunya), tapi biar cepet lewat mana ya kalau dari rumahku? Oh iya kan ada algoritma yang bisa nentuin itu gak sih?

Nah aku terlalu malas, jadi sekarang tugas kalian adalah ngebuat program yang bisa nentuin jalan tercepat dari rumah yang dipilih dari peta yang nanti aku kasih, mau kan kalian ngebuat ini untukku 😬

Constraints:

- $0 \leq A, B, N, S, v1, v2, \text{weight} < \text{LONG_MAX}$ (2147483647)

Input:

- **Baris pertama berisi:**

1. A, yaitu jumlah Vertex dalam Graph
 2. B, yaitu jumlah Edge dalam Graph
- **Baris sejumlah B berikutnya akan berisi:** v1, v2, dan weight, yaitu vertex yang akan dihubungkan dan berat edge tersebut.
 - **Baris selanjutnya berisi:** N, yaitu jumlah operasi mencari path yang akan dilakukan
 - Baris sejumlah N berikutnya berisi : S, yaitu vertex asal dari semua path yang akan dibuat. S dinilai tidak valid jika vertex tersebut tidak ada dalam graph.

Output:

- Output yang akan dihasilkan adalah semua path terpendek dari S menuju semua vertex dalam graph kecuali vertex S itu sendiri. Dengan contoh format adalah

```
---From [S(vertexSource)] to [vertexDestination]---
```

Dan path di print dengan urutan vertex 0 ke A-1 dengan contoh format

```
source - a - b - c - destination
```

- Jika S tidak valid, maka print **Invalid starting point**

Examples :

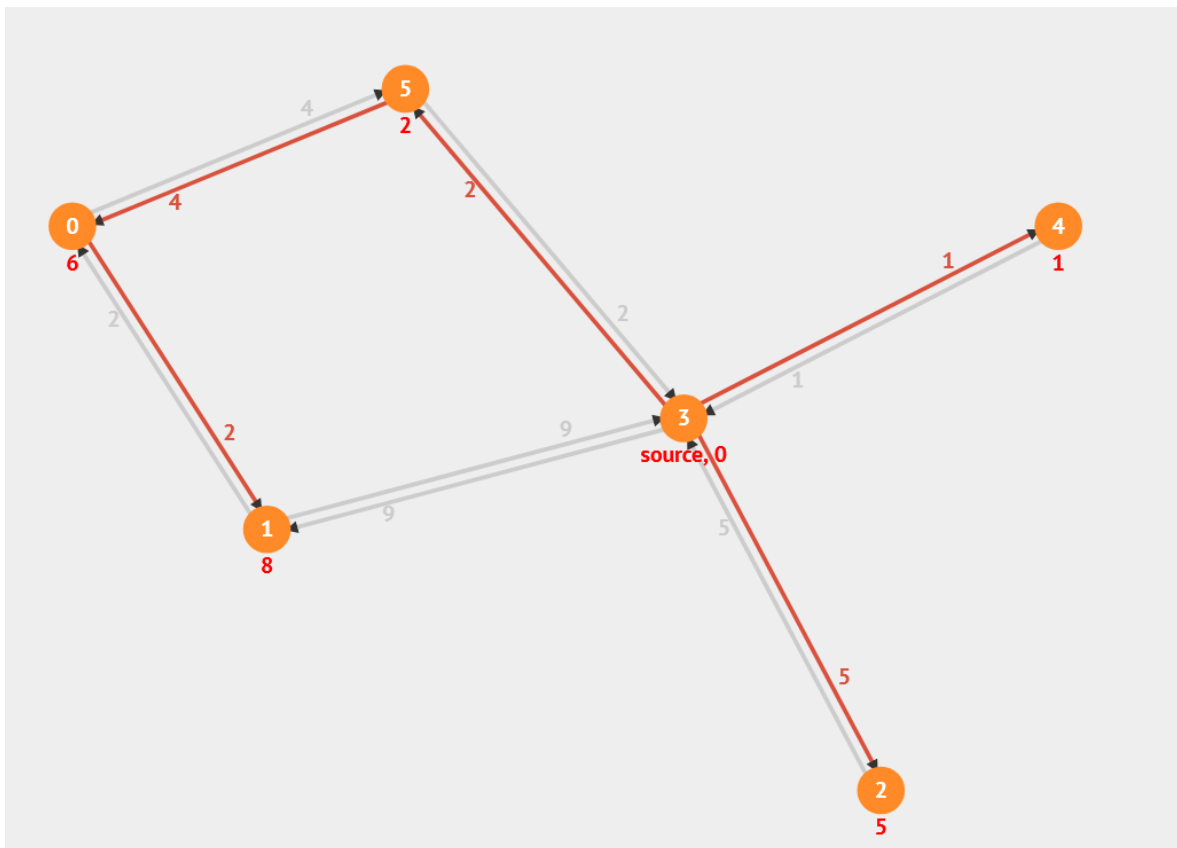
- **Example 1**

Input
6 6
0 1 2
0 5 4
1 3 9
5 3 2
3 4 1
3 2 5
2
3
6
Output
---From 3 to 0---
3 - 5 - 0
---From 3 to 1---

```
3 - 5 - 0 - 1
---From 3 to 2---
3 - 2
---From 3 to 4---
3 - 4
---From 3 to 5---
3 - 5
Invalid starting point
```

Explanation:

S : 3



Disini bisa terlihat graph yang dihasilkan oleh input diatas. Dengan menggunakan algoritma shortest path, kita dapat menentukan rute paling pender dengan asal S : 3.

Soal ini disponsori oleh Global Engagement : Jadilah Buddy Department untuk menambah teman internasional!

See You Soon!

By R. Rafif Aqil Aabid Hermawan

Time Limit : 1s



Description :

It's not a Goodbye it's a See you Soon!

So I would like to surprise my friends before they all go abroad (They all are going to do Double Degree 🤖). I already have a map with my friend's location, but what is the fastest way to visit my friends from my home? Oh that's right, I have the algorithm that can determine that. Why don't I use that?

But I'm too lazy, so now it's your job to create a program that can determine the fastest path from my chosen home (I have a lot) from the map I will give (In graph form of course). Will you help me pleaseeeee 🙏

Constraints:

- $0 \leq A, B, N, S, v1, v2, \text{weight} < \text{LONG_MAX}$ (2147483647)

Input:

- **First line contains:**

3. A, The amount of vertex in the graph
 4. B, The amount of edge in the graph
- **Next B line contains:** v1, v2, dan weight, the vertex that will be connected and the associated weight of the edge.
 - **Next line contains :** N, the amount of operation of path finding.
 - **Next N line contains :** S, that is the source vertex from all path that will be made. S is not valid when the vertex doesn't exist in the graph.

Output:

- Output that will be produced is all shortest path from S to all vertex in a graph except S itself. Example of the output is

---From [S(vertexSource)] to [vertexDestination]---

And the path is printed with the order from 0 to A-1 with the format example

source - a - b - c - destination

- If S isn't valid, print Invalid starting point

Examples :

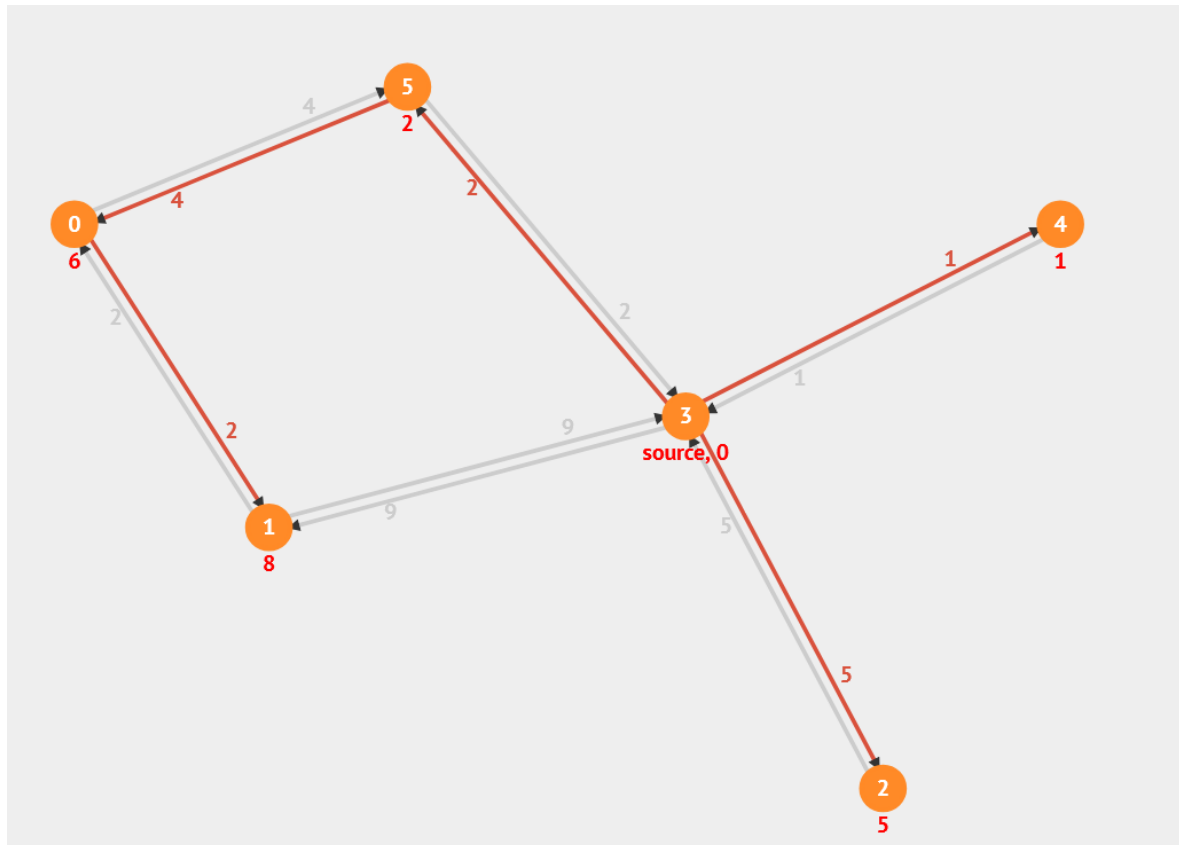
- **Example 1**

Input
6 6
0 1 2
0 5 4
1 3 9
5 3 2
3 4 1
3 2 5
2
3
6
Output
---From 3 to 0---
3 - 5 - 0
---From 3 to 1---
3 - 5 - 0 - 1
---From 3 to 2---
3 - 2
---From 3 to 4---

```
3 - 4
---From 3 to 5---
3 - 5
Invalid starting point
```

Explanation:

S : 3



Here we can see the graph generated from the input above and the shortest path from the source to all vertex in the graph. Using the shortest path algorithm, we can get the shortest route from S : 3.

This question is sponsored by Global Engagement : Become a Buddy Department and gain international friend!