

And It's Going Dutch

Author: Farel Hanif Andaru

Time limit	1 s
Memory Limit	256 MB



Deskripsi

Anda sebagai race engineer dari Max Verstappen harus membantu max menentukan keputusan yang akan dilakukan pada **salah satu tikungan**, dengan beberapa data racing line didapatkan dari practice session yang berupa weighted graph. Anda harus memutuskan max akan menggunakan racing line yang mana dari sekian banyaknya pilihan untuk mendahului rival nya Oscar Piastri. Maximum_weight adalah batas weight yang dapat dimiliki oleh max untuk mendahului Oscar.

Untuk racing line yang berisiko anda harus melakukan perhitungan weight dari setiap edges racing line berisiko tersebut dikarenakan dive bombing sangat berisiko maka max tidak melakukan manuver tersebut pada sesi practice.

Karena Max Verstappen berasal dari belanda maka dia ingin anda untuk **menggunakan algoritma dijkstra** yang pembuatnya juga berasal dari belanda.

Rumus Weight Edge Berisiko

$$\text{Weight} = \text{weight_prev} + \text{weight_start}$$

weight_prev adalah weight dari edge sebelum edge ini

weight_start adalah weight dari edge paling awal

Batasan

$$2 < n_vertices < 100$$

$$1 < n_Edges < 99$$

$$1 < \text{Weight} < 1000$$

$$0 \leq \text{isRisky} \leq 1$$

$$1 < \text{maximum_weight} < 100000$$

Modifikasi Algoritma Dijkstra ini: [Dijkstra by Alpro](#)

Input

```
n_vertices  
n_edges  
vertex1 vertex2 weight isRisky  
start_node  
finish_node  
maximum_weight
```

Note: weight edge risky pada saat input diisi dengan 0

Output

- Jika Max **berhasil** mendahului Oscar menggunakan **racing line berisiko** maka keluarkan "Press to overtake and dive bomb please max"
- Jika Max **berhasil** mendahului Oscar menggunakan **racing line normal** maka keluarkan
"Decrease lift and coast please max, oscar is slower in front of you"
- Jika **tidak ada** racing line yang memungkinkan Max untuk mendahului Oscar maka keluarkan "Increase lift and coast please max, just try to stay in front of Lando"

Contoh

Input:

```
8
8
0 1 10 0
1 2 25 0
2 3 35 0
3 4 48 0
4 5 57 0
2 6 0 1
6 7 0 1
7 5 0 1
0
5
200
```

Output:

```
Press to overtake and dive bomb please max
```

hint:

- Max sangat percaya dengan strategi yang anda pilih, sehingga max tidak akan mengganti racing line setelah berbelok untuk pertama kalinya sampai tikungan tersebut selesai
- Vertex Start pasti hanya mempunyai 1 edge jadi jangan terlalu dipusingkan mencari weight_start

Penjelasan ada di halaman selanjutnya!

Mercedes not happy, red bull will be delighted, they have shared a brilliant championship battle but the championship could only won by one **and it's going dutch** in 2021 MAX VERSTAPPEN FOR THE FIRST TIME EVER IS THE CHAMPION OF THE WORLD

Selamat mengerjakan rek, intinya jangan di-overthink 😊, glhf



Sesuai input, diketahui bahwa edge yang berwarna hijau adalah edge normal dan sudah memiliki weight semua, sedangkan edge berwarna merah putus putus tidak mempunyai weight dan kita harus menghitung berapa weight dari edge tersebut.

"0 1 10 0" berarti edge antara vertex 0 dan 1 memiliki weight 10 dan tidak risky.

"2 6 0 1" berarti edge antara vertex 2 dan 6 risky dan tidak diketahui weight nya.

Untuk mendapatkan weight dari edge 2-6 adalah:

$\text{weight} = \text{weight_prev} + \text{weight_start}$

$\text{weight} = \text{weight edge 12} + \text{weight 01}$

And It's Going Dutch

Author: Farel Hanif Andaru

Time limit	1 s
Memory Limit	256 MB



Description

As Max Verstappen's race engineer, you must help Max decide on the next course of action at each corner. Using racing line data obtained from the practice session, which is in the form of a weighted graph, you need to determine whether Max should use the normal racing line or a risky racing line to overtake his rival, Oscar Piastri. Maximum_weight represents the limit on Max's weight for him to be ahead of Oscar.

For the risky racing line, you will need to calculate the weight of each of its edges. This is because dive-bombing is very risky, so Max did not attempt this maneuver during the practice session.

Since Max Verstappen is from the Netherlands, he wants you to **use Dijkstra's algorithm**, as its creator is also from the Netherlands.

Formula to Calculate Risky Edge Weight

$$\text{Weight} = \text{weight_prev} + \text{weight_start}$$

weight_prev is the weight of the previous edge (or the edge before this one).

weight_start is the weight of the initial edge (or the very first edge / the starting edge).

Constrains

$2 < n_vertices < 100$

$1 < n_Edges < 99$

$1 < Weight < 1000$

$0 \leq isRisky \leq 1$

$1 < maximum_weight < 100000$

Modify this Dijkstra algorithm provided by Alpro: [Dijkstra by Alpro](#)

Input

```
n vertices
n_edges
vertex1 vertex2 weight isRisky
start_node
finish_node
maximum_weight
```

Note: When inputting, the risky edge weight is filled with 0

Output

- If Max **successfully** overtakes Oscar using the **risky racing line**, then output: "Press to overtake and dive bomb please max"
- If Max **successfully** overtakes Oscar using the **normal racing line**, then output: "Decrease lift and coast please max, oscar is slower in front of you"
- If there is **no racing** line that allows Max to overtake Oscar, then output "Increase lift and coast please max, just try to stay in front of Lando"

Example

Input:

```
8
8
0 1 10 0
1 2 25 0
2 3 35 0
3 4 48 0
4 5 57 0
2 6 0 1
6 7 0 1
7 5 0 1
0
5
200
```

Output:

```
Press to overtake and dive bomb please max
```

hint:

- Max is so confident in your chosen strategy that once he makes his initial turn, he won't alter his racing line until he's through the corner.
- The Start Vertex will definitely only have one edge, so don't worry too much about finding weight_start.

Explanation on the next page!

Mercedes not happy, red bull will be delighted, they have shared a brilliant championship battle but the championship could only won by one **and it's going dutch** in 2021 MAX VERSTAPPEN FOR THE FIRST TIME EVER IS THE CHAMPION OF THE WORLD

Selamat mengerjakan rek, intinya jangan di-overthink 😊, glhf



As per the input, it's known that green edges are normal edges and already have all their weights. In contrast, red dashed edges do not have a weight, and we must calculate the weight of these edges.

"0 1 10 0" means the edge between vertex 0 and vertex 1 has a weight of 10 and is not risky.

"2 6 0 1" means the edge between vertex 2 and vertex 6 is risky, and its weight is unknown.

To get the weight of edge 2-6:

$\text{weight} = \text{weight_prev} + \text{weight_start}$

$\text{weight} = \text{weight of edge 1-2} + \text{weight of edge 0-1}$