

# Adatszerkezetek és algoritmusok 1. kis házi feladat

## Csatornatisztítás

Marvin egy csatornatisztító robot, aki minden reggel bejárja a város alatti csatornarendszert, és letisztítja a falakat. Írj programot, amely a csatornahálózat térképének függvényében meghatároz egy olyan útvonalat, amely az egészet bejárja.

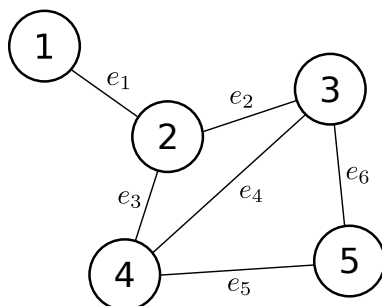
Marvin négy keréken gurul, és a jobb oldalán van a tisztító eszköz, tehát mindig a mindenkori jobboldali falat tisztítja; tehát minden folyosón kétszer kell végigmenni, egyszer az egyik, következőnek a másik irányba.

Viszont Marvin utálja a felesleges munkát, ezért nem akar egy olyan folyosón olyan falat takarítani, amelyet már letakarított. Ezért a programodnak olyan útvonalat kell terveznie, amely minden folyosón pontosan kétszer halad át: egyszer az egyik, következőnek a másik irányba.

A csatornarendszer  $M$  folyosóból és a találkozási pontokon kereszteződésekből áll. Az  $N$  darab kereszteződést 1-től  $N$ -ig számozzuk. A térképen fel vannak sorolva a folyosók, két számmal reprezentálva:  $(a, b)$  folyosó köti össze az  $a$ -val számozott kereszteződést a  $b$ -vel számozott kereszteződéssel. A csatornarendszer mindig összefüggő, tehát bármely két kereszteződés között van legalább egy útvonal.

Marvin bármelyik kereszteződésből elindulhat, és bárhova megérkezhet (nem feltétlenül ugyanoda).

## Példa



1. ábra. Példa gráf.

Példa bemenet (lerajzolva az 1. ábrán):  $N = 5$ ,  $M = 6$ , az élek:

1 2  
2 3  
2 4  
4 3  
4 5  
3 5

Egy lehetséges megoldás:

$2 \xrightarrow{e_2} 3 \xrightarrow{e_4} 4 \xrightarrow{e_3} 2 \xrightarrow{e_1} 1 \xrightarrow{\bar{e}_1} 2 \xrightarrow{\bar{e}_3} 4 \xrightarrow{e_5} 5 \xrightarrow{e_6} 3 \xrightarrow{\bar{e}_6} 5 \xrightarrow{\bar{e}_5} 4 \xrightarrow{\bar{e}_4} 3 \xrightarrow{\bar{e}_2} 2$

Itt a nyilak feletti feliratok az éleket jelzik (az 1. ábra szerinti jelöléssel), az oda- és visszafele irányt önkényesen megkülönböztetve felülvonással. A te implementációd egyszerűen csak a számsorozatot fogja visszaadni.

Nem ez az egyetlen megoldás.

## Limitek

A feladatok során a bemenetek nem léphetnek túl a limiteken, ezt nem kell külön ellenőrizni.

- $2 \leq N \leq 10^6$
- $2 \leq M \leq 10^7$
- Minden folyosóra:  $1 \leq a, b \leq N$
- Időlimit: tesztetesenként 5 másodperc
- Memórialimit:
  - heap: 500 MiB (nem számítva az ellenőrző által lefoglalt,  $\approx 120$  MB-t)
  - stack: 8 MiB

## API

A megoldást a következő függvényben kell implementálni:

```
template<typename Iterator>
std::vector<std::int32_t> plan(Iterator corridors_begin, Iterator corridors_end, size_t N);
```

A típusparaméter egy iterátor típusa<sup>1</sup>. Az iterátor által mutatott érték típusa Corridor:

```
struct Corridor {
    std::int32_t from, to;
};
```

A megoldás egy `std::vector<int>`, amely az útvonal során érintett kereszteződések sorszámát tartalmazza sorrendben, beleértve azt, ahonnan Marvin elindul, és azt, ahova érkezik.

## Tanácsok

Próbáld meg először lerajzolni a teszt eseteket (az egyszerűbbekkel kezdve), és fejből kitalálni, hogy mi lehet egy jó megoldás. Próbáld elemezni, hogy hogyan néznek ki a megoldások. Ha van egy működőnek látszó algoritmusod, próbáld meg olyan esetet rajzolni, ahol nem működik.

Másik fontos kérdés, hogy hogyan reprezentáljátok a csatornahálózatot?

Az mindenképp segít, ha a matekos tárgyaktól tanultakat nem felejtet el.

*Jó munkát!*

---

<sup>1</sup>A behelyettesített paraméter egy *LegacyContiguousIterator* (szintén nem kell ellenőrizni).