

## Sípálya

Egy hegyoldalon lévő sípályán minden választható lesiklási útvonal a start ponttól a célpontig vezet. A pálya elágazási pontokat tartalmaz, csak ilyen pontokon lehet a lesiklás során pályaszakaszt váltani. Két elágazási pont közötti részt nevezzük pályaszakasznak. Minden verseny előtt a szervezők ellenőrzik a pálya alkalmasságát. Ebből a célból ellenőrök mennek végig a pályán. Minden ellenőr a start ponttól indul és végig megy egy útvonalon a célpontig. Az a cél, hogy a legkevesebb ellenőrt kelljen alkalmazni ahhoz, hogy minden pályaszakaszon áthaladjon legalább egy ellenőr.

A pálya olyan, hogy a térképe lerajzolható úgy, hogy bármely két pályaszakasz nem metszi egymást (nincs sem alagút, sem felüljáró), csak elágazási pontban lehet közös pontjuk. Két elágazási pont között legfeljebb egy pályaszakasz van. Minden pályaszakaszon csak egy irányban lehet lesiklani, a magasabban lévő pontból az alacsonyabban lévő pont felé. Bármely elágazási pont elérhető a start pontból és bármely elágazási pontból el lehet jutni a célpontba.

Készíts programot, amely kiszámítja, hogy legkevesebb hány ellenőrt kell alkalmazni!

### Bemenet

A *standard bemenet* első sorában az elágazási pontok száma ( $1 \leq N \leq 20\,000$ ) van. Az elágazási pontokat az  $1, \dots, N$  számokkal azonosítjuk, a start pont azonosítója az 1, a célponté  $N$ . A további  $N-1$  sor közül az  $i$ -edik az  $i$ -edik elágazási pontból induló pályaszakaszokat adja meg Nyugatról Kelet felé haladó sorrendben. A sorban az első szám a pontból induló pályaszakaszok száma, ezt követik azon elágazási pontok sorszámai, ahova a pályaszakasz vezet.

### Kimenet

A *standard kimenet* első és egyetlen sorába egy egész számot kell írni, az alkalmazandó legkevesebb ellenőr számát!

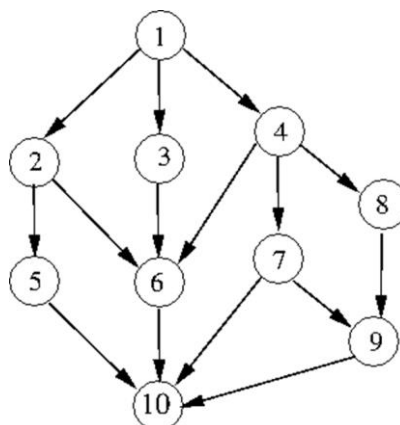
### Példa

bemenet

```
10
3 2 3 4
2 5 6
1 6
3 6 7 8
1 10
1 10
2 10 9
1 9
1 10
```

kimenet

7



### Korlátok

Időlimit: 0.25 mp.

Memórialimit: 32 MB

A pontok 20%-szerezhető olyan bemenetekre, ahol  $N \leq 200$ .