Keys

De architect Timothy heeft een nieuw ontsnappingsspel bedacht. In dit spel zijn er n kamers genummerd van 0 tot en met $n-1$. In het begin bevat elke kamer precies één sleutel. Elke sleutel heeft een type, wat een integer is van 0 tot en met $n-1$. Het type van de sleutel in kamer i ($0 \leq i \leq n-1$) is $r[i]$. Merk op dat meerdere kamers sleutels van hetzelfde type kunnen bevatten, oftewel, de waarden $r[i]$ zijn niet noodzakelijk allemaal verschillend.

Er zijn ook m **bidirectionele** verbindingen in het spel, genummerd van 0 tot en met $m-1$. Verbinding j ($0 \leq j \leq m-1$) verbindt een tweetal verschillende kamers $u[j]$ en $v[j]$. Een tweetal kamers kan door meerdere verbindingen verbonden zijn.

Het spel wordt door één speler gespeeld, die sleutels verzamelt en zich door de kamers verplaatst via de verbindingen. We zeggen dat een speler **via** verbinding j loopt als ze deze verbinding gebruiken om zich te verplaatsen van kamer $u[j]$ naar kamer $v[j]$, of andersom. De speler kan alleen via verbinding j lopen als ze een sleutel van type $c[j]$ al eerder verzameld hebben.

Op elk punt van het spel is de speler in een kamer x en kan twee soorten acties uitvoeren:

- pak de sleutel in kamer x, waarvan het type $r[x]$ is (tenzij ze deze al eerder gepakt hebben).
- loop via verbinding j, waarbij óf $u[j] = x$ óf $v[j] = x$, als de speler al eerder een sleutel van type $c[j]$ gepakt heeft. Merk op dat de speler een eenmaal gepakte sleutel **nooit** weggooit.

De speler **begint** het spel in een kamer s zonder dat ze een sleutel in bezit hebben. Een kamer t is **bereikbaar** vanuit kamer s, als de speler die start in kamer s een aantal acties kan uitvoeren zoals hierboven beschreven om vervolgens in kamer t uit te komen.

Noem $p[i]$ het aantal kamers dat je vanuit kamer i ($0 \leq i \leq n-1$) kunt bereiken. Timothy wil de verzameling indices i weten waarvoor $p[i]$ minimaal is voor alle $0 \leq i \leq n-1$.

Implementatiedetails

Implementeer de volgende functie:

```
int[] find_reachable(int[] r, int[] u, int[] v, int[] c)
```

- r: een array van lengte n. Voor elke i ($0 \leq i \leq n-1$) geldt dat de sleutel in kamer i is van het type $r[i]$.
- u, v: twee arrays van lengte m. Voor elke j ($0 \leq j \leq m-1$) geldt dat verbinding j de kamers $u[j]$ en $v[j]$ verbindt.
- c: een array van lengte m. Voor elke j ($0 \leq j \leq m-1$) geldt dat $c[j]$ het type sleutel is dat nodig is om via verbinding j te lopen.

- Deze functie moet een array a van lengte n teruggeven. Voor elke $0 \leq i \leq n-1$, moet de waarde van $s[i]$ gelijk zijn aan 1 als voor elke j met $0 \leq j \leq n-1$ geldt $p[i] \leq p[j]$. Anders moet de waarde van $s[i]$ gelijk aan 0 zijn.

Voorbeelden

Voorbeeld 1

Neem de volgende aanroep:

```
find_reachable([0, 1, 1, 2],
               [0, 0, 1, 1, 3], [1, 2, 2, 3, 1], [0, 0, 1, 0, 2])
```

Als de speler in kamer 0 begint, kunnen ze de volgende rij acties doen:

Huidige kamer	Actie
0	Verzamel sleutel van type 0
0	Loop via verbinding 0 naar kamer 1
1	Verzamel sleutel van type 1
1	Loop via verbinding 2 naar kamer 2
2	Loop via verbinding 2 naar kamer 1
1	Loop via verbinding 3 naar kamer 3

Dus kamer 3 is bereikbaar vanuit kamer 0. Op een vergelijkbare manier kunnen we rijen van acties maken om te laten zien dat vanuit kamer 0 alle kamers bereikbaar zijn, dus $p[0] = 4$. De tabel hieronder laat de bereikbare kamers zien voor alle begin kamers:

Begin kamer i	Bereikbare kamers	$p[i]$
0	$[0, 1, 2, 3]$	4
1	$[1, 2]$	2
2	$[1, 2]$	2
3	$[1, 2, 3]$	3

De kleinste waarde van $p[i]$ over alle kamers is 2 en deze wordt aangenomen voor $i = 1$ en $i = 2$. Dus de functie zou $[0, 1, 1, 0]$ moeten teruggeven.

Voorbeeld 2

```
find_reachable([0, 1, 1, 2, 2, 1, 2],
               [0, 0, 1, 1, 2, 3, 3, 4, 4, 5],
               [1, 2, 2, 3, 3, 4, 5, 5, 6, 6],
               [0, 0, 1, 0, 0, 1, 2, 0, 2, 1])
```

De tabel hieronder toont de bereikbare kamers:

Begin kamer i	Bereikbare kamers	$p[i]$
0	$[0, 1, 2, 3, 4, 5, 6]$	7
1	$[1, 2]$	2
2	$[1, 2]$	2
3	$[3, 4, 5, 6]$	4
4	$[4, 6]$	2
5	$[3, 4, 5, 6]$	4
6	$[4, 6]$	2

De kleinste waarde van $p[i]$ over alle kamers is 2, en deze wordt aangenomen voor $i \in \{1, 2, 4, 6\}$. Dus, geeft deze functie $[0, 1, 1, 0, 1, 0, 1]$ terug.

Voorbeeld 3

```
find_reachable([0, 0, 0], [0], [1], [0])
```

De volgende tabel toont de bereikbare kamers:

Begin kamer i	Bereikbare kamers	$p[i]$
0	$[0, 1]$	2
1	$[0, 1]$	2
2	$[2]$	1

De kleinste waarde van $p[i]$ over alle kamers is 1, en deze wordt aangenomen voor $i = 2$. Dus, geeft deze functie $[0, 0, 1]$ terug.

Randvoorwaarden

- $2 \leq n \leq 300\,000$
- $1 \leq m \leq 300\,000$
- $0 \leq r[i] \leq n - 1$ voor elke $0 \leq i \leq n - 1$
- $0 \leq u[j], v[j] \leq n - 1$ en $u[j] \neq v[j]$ voor elke $0 \leq j \leq m - 1$

- $0 \leq c[j] \leq n-1$ voor elke $0 \leq j \leq m-1$

Subtaken

1. (9 punten) $c[j] = 0$ voor alle $0 \leq j \leq m-1$ en $n, m \leq 200$
2. (11 punten) $n, m \leq 200$
3. (17 punten) $n, m \leq 2000$
4. (30 punten) $c[j] \leq 29$ (voor alle $0 \leq j \leq m-1$) en $r[i] \leq 29$ (voor alle $0 \leq i \leq n-1$)
5. (33 punten) Geen verdere randvoorwaarden.

Voorbeeldgrader

De voorbeeldgrader leest de invoer in het volgende formaat:

- regel 1: n m
- regel 2: $r[0]$ $r[1]$ $\ldots$ $r[n-1]$
- regel $3+j$ ($0 \leq j \leq m-1$): $u[j]$ $v[j]$ $c[j]$

De voorbeeldgrader schrijft de waarde teruggekregen van `find_reachable` in het volgende formaat:

- regel 1: $a[0]$ $a[1]$ $\ldots$ $a[n-1]$