

## Сфинкстин Табышмагы

Улуу Сфинкстин силер үчүн табышмактары бар. Сизге  $N$  чокулар менен граф берилет. Чокулар 0 дөн  $N - 1$  ге чейин номерленген. Графта 0 дөн  $M - 1$  ге чейин номерленген  $M$  кырлары бар. Ар бир кыры ар түрлүү чокулардын жуптарын бириктирет жана эки багыттуу. Тактап айтканда, ар бир  $j$  үчүн 0 дөн  $M - 1$  ге чейин  $j$  кыры  $X[j]$  жана  $Y[j]$  чокуларын бириктирет. Ар кандай жуп чокуларды бириктирүүчү эң көп дегенде бир кыр бар. Эгерде эки чоку бир кыр менен туташтырылса алар **кошуна** деп аталат.

Эгерде  $v_0, v_1, \dots, v_k$  ( $k \geq 0$  үчүн) чокуларынын ырааттуулугунда ар бир ырааттуу эки чоку кошуна болушса, ырааттуулук **жол** деп аталат. Мындай  $v_0, v_1, \dots, v_k$  ырааттуулугу  $v_0$  жана  $v_k$  чокуларын **туташтырат** деп айтабыз. Сизге берилген графта чокулардын ар бир жубу кандайдыр бир жол менен байланышкан.

0 дөн  $N$  ге чейин номерленген  $N + 1$  түстөр бар.  $N$  түсү өзгөчө жана **Сфинкстин түсү** деп аталат. Ар бир чокуга түс ыйгарылган. Тактап айтканда,  $i$  ( $0 \leq i < N$ ) чокусунун түсү  $C[i]$ . Бир нече чокулар бирдей түстө болушу мүмкүн жана эч бир чокуга ыйгарылбаган түстөр болушу мүмкүн. Эч бир чокуда Сфинкстин түсү жок, башкача айтканда,  $0 \leq C[i] < N$  ( $0 \leq i < N$ ).

Эгерде  $v_0, v_1, \dots, v_k$  ( $k \geq 0$  үчүн) жолдун бардык чокулары бирдей түстө болсо, жол **монохромдук** деп аталат, б.а.  $C[v_l] = C[v_{l+1}]$  (ар бир  $l$  ( $0 \leq l < k$ ) үчүн). Кошумчалай кетсек, эгерде  $p$  жана  $q$  ( $0 \leq p < N$ ,  $0 \leq q < N$ ) чокулары монохромдук жол менен байланышкан гана болсо алар бирдей **монохромдук компонентте**.

Сиз чокуларды жана кырларды билесиз, бирок ар бир чокуда кандай түс бар экенин билбейсиз. Сиз **кайра боёо эксперименттерин** аткаруу менен чокулардын түстөрүн билгиңиз келет.

Кайра боёо экспериментинде сиз чокулардын каалаган санын боёсоңуз болот. Тактап айтканда, кайра боёо экспериментин аткаруу үчүн сиз адегенде узундугу  $N$  болгон  $E$  массивин тандайсыз, мында ар бир  $i$  ( $0 \leq i < N$ ) үчүн  $E[i]$ ,  $-1$  жана  $N$  дин ортосунда ( $-1$  жана  $N$  ди **кошкондо**). Андан кийин, ар бир  $i$  чокусунун түсү  $S[i]$  болуп калат, мында  $S[i]$ :

- $C[i]$ , бул  $i$  дын баштапкы түсү, эгерде  $E[i] = -1$  болсо, же
- $E[i]$ , башка учурларда.

Мында сиз Сфинкстин түсүн кайра боёоңузда колдоно аласыз дегенди билдирерин эске алыңыз.

Акырында, Улуу Сфинкс ар бир  $i$  ( $0 \leq i < N$ ) чокусунун түсүн  $S[i]$  деп койгондон кийин, графтагы монохромдук компоненттердин санын жарыялайт. Жаңы боёо ушул өзгөчө кайра боёо эксперименти үчүн гана колдонулат, андыктан **баардык чокулардын түстөрү эксперимент аяктагандан кийин баштапкы түстөрүнө кайтып келет**.

Сиздин милдетиңиз эң көп 2 750 боёо эксперименттерин аткаруу менен графтын чокуларынын түстөрүн аныктоо. Эгер сиз ар бир кошуна чокулардын түсү бирдей экендигин туура аныктасаңыз, жарым-жартылай упай ала аласыз.

## Ишке ашыруу чоо-жайы

Сиз төмөнкү процедурасын ишке ашырууңуз керек.

```
std::vector<int> find_colours(int N,
                             std::vector<int> X, std::vector<int> Y)
```

- $N$  : графтагы чокулардын саны.
- $X, Y$  : кырларды сүрөттөгөн  $M$  узундуктагы массивдер.
- Бул процедура графтагы чокулардын түстөрүн билдирген узундугу  $N$  болгон  $G$  массивин кайтарышы керек.
- Бул процедура ар бир сыноо учуру үчүн бир жолу чакырылат.

Жогорудагы процедура кайра боёо эксперименттерди аткаруу үчүн төмөнкү процедураны чакыра алат:

```
int perform_experiment(std::vector<int> E)
```

- $E$  : чокуларды кантип кайра боёо керектигин көрсөткөн  $N$  узундуктагы массив.
- Бул процедура  $E$  ылайык чокуларды кайра боёгондон кийин монохромдук компоненттердин санын кайтарат.
- Бул процедураны эң көп 2 750 жолу чакырса болот.

Грейдер **адаптивдүү эмес**, башкача айтканда, чокулардын түстөрү `find_colours` чакырыгы жасалганга чейин бекитилет.

## Чектөөлөр

- $2 \leq N \leq 250$
- $N - 1 \leq M \leq \frac{N \cdot (N - 1)}{2}$
- $0 \leq X[j] < Y[j] < N$  ар бир  $j$  ( $0 \leq j < M$ ) үчүн.

- $X[j] \neq X[k]$  же  $Y[j] \neq Y[k]$  ар бир  $j$  жана  $k$  ( $0 \leq j < k < M$ ) үчүн.
- Ар бир жуп чокулар кандайдыр бир жол менен байланышкан.
- $0 \leq C[i] < N$  ар бир  $i$  ( $0 \leq i < N$ ) үчүн.

## Кошумча тапшырмалар

Кошумча тапшырмалар	Упай	Кошумча чектөөлөр
1	3	$N = 2$
2	7	$N \leq 50$
3	33	Граф бул жол: $M = N - 1$ жана $j$ менен $j + 1$ чокулары кошуналар ( $0 \leq j < M$ ).
4	21	Граф толук: $M = \frac{N \cdot (N-1)}{2}$ жана ар бир жуп чокусу кошуна.
5	36	Эч кандай кошумча чектөөлөр.

Ар бир кошумча тапшырмада, эгерде сиздин программаңыз кошуна чокулардын ар бир жуп үчүн алардын түсү бирдей экендигин туура аныктаса, жарым-жартылай упай ала аласыз.

Тагыраак айтканда, эгер бардык сыноо учурларында `find_colours` кайтарып берген  $G$  массиви  $C$  массивине так окшош болсо кошумча тапшырманын толук упайларын аласыз (б.а.  $G[i] = C[i]$  бардык  $i$  ( $0 \leq i < N$ ) үчүн). Болбосо, бардык сыноо учурларында төмөнкү шарттар аткарылса, кошумча тапшырманын 50% упайын аласыз:

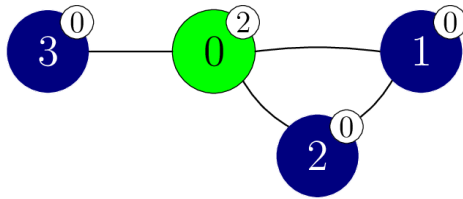
- $0 \leq G[i] < N$  ар бир  $i$  ( $0 \leq i < N$ ) үчүн;
- Ар бир  $j$  ( $0 \leq j < M$ ) үчүн:
  - эгерде  $G[X[j]] = G[Y[j]]$  болсо гана  $C[X[j]] = C[Y[j]]$  болот.

## Мисал

Төмөнкү чакырыкты карап көрөлү.

```
find_colours(4, [0, 1, 0, 0], [1, 2, 2, 3])
```

Бул мисал үчүн чокулардын (жашыруун) түстөрү  $C = [2, 0, 0, 0]$  менен берилген дейли. Бул сценарий төмөнкү сүрөттө көрсөтүлгөн. Түстөр кошумча түрдө ар бир чокуга тиркелген ак белгидеги сандар менен көрсөтүлөт.



Бул процедура `perform_experiment` процедурасын төмөнкүдөй чакырышы мүмкүн.

```
perform_experiment([-1, -1, -1, -1])
```

Бул чалууда эч бир чоку кайра боёлбойт, анткени бардык чокулар баштапкы түстөрүн сактап калат.

1 чокусун жана 2 чокусун карап көрөлү. Экөөнүн тең түсү 0 жана 1, 2 жолу монохромдук жол. Натыйжада, 1 жана 2 чокулары бирдей монохромдук компонентте.

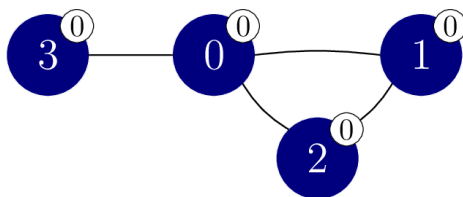
1 чокусун жана 3 чокусун карап көрөлү. Экөөнүн тең түсү 0 болсо да, аларды бириктирген монохромдук жол жок болгондуктан, алар ар кандай монохромдук компоненттерде.

Жалпысынан 3 монохроматтык компоненттери бар, чокулары  $\{0\}$ ,  $\{1,2\}$  жана  $\{3\}$ . Ошентип, бул чакыруу 3 кайтарып берет.

Эми процедура `perform_experiment` деп төмөнкүдөй чакырылышы мүмкүн.

```
perform_experiment([0, -1, -1, -1])
```

Бул чакырууда 0 чокусу гана 0 түсүнө кайра боёлуп, анын натыйжасында төмөнкү сүрөттө көрсөтүлгөн боёо болот.

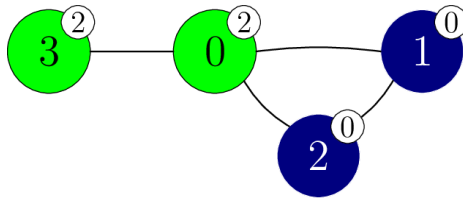


Бул чалуу 1 кайтарат, анткени бардык чокулар бир монохроматтык компонентке таандык. Эми 1, 2 жана 3 чокуларынын 0 түстө экенин чыгара алабыз.

Андан кийин процедура төмөнкүдөй `perform_experiment` процедурасын чакырышы мүмкүн.

```
perform_experiment([-1, -1, -1, 2])
```

Бул чакырууда 3 чокусу 2 түсүнө кайра боёлуп, анын натыйжасында төмөнкү сүрөттө көрсөтүлгөн боёо пайда болот.



Бул чалуу 2 кайтарат, анткени 2 монохромдук компоненттер бар, алардын чокулары  $\{0, 3\}$  жана  $\{1, 2\}$ . 0 чокусунда 2 түстө экенин биле алабыз.

Андан кийин `find_colours` процедурасы  $[2, 0, 0, 0]$  массивин кайтарат.  $C = [2, 0, 0, 0]$  болгондуктан, толук балл берилет.

Эскертүү, ошондой эле бир нече кайтаруу боёолору бар, булар үчүн 50% упай берилмек, мисалы  $[1, 2, 2, 2]$  же  $[1, 2, 2, 3]$ .

## Үлгү Грейдер

Киргизүү форматы:

```
N M
C[0] C[1] ... C[N-1]
X[0] Y[0]
X[1] Y[1]
...
X[M-1] Y[M-1]
```

Чыгуу форматы:

```
L Q
G[0] G[1] ... G[L-1]
```

Бул жерде,  $L$  `find_colours` тарабынан кайтарылган  $G$  массивинин узундугу жана  $Q$  - `perform_experiment` үчүн чакыруулардын саны.