



Split the Attractions

Existem n atrações em Baku, numeradas de 0 a $n - 1$. Existem também m estradas nos dois sentidos, numeradas de 0 a $m - 1$. Cada estrada liga duas atrações diferentes. É possível viajar entre qualquer par de atrações usando as estradas.

Fatima está a planear visitar todas as atrações em três dias. Ela já decidiu que vai visitar a atrações no primeiro dia, b atrações no segundo dia e c atrações no terceiro dia. Por isso mesmo, ela quer particionar as n atrações em três conjuntos A , B e C com tamanhos a , b e c , respetivamente. Cada atração deverá pertencer a exatamente um conjunto e portanto $a + b + c = n$.

Fatima gostaria de descobrir os conjuntos A , B e C de tal modo que **pelo menos dois** dos três conjuntos sejam **conexos**. Um conjunto de S atrações é chamado de conexo se for possível viajar entre qualquer par de atrações de S usando as estradas e sem passar por qualquer atração que não esteja em S . A partição das atrações nos conjuntos A , B e C é chamada de **válida** se satisfizer as condições descritas anteriormente.

Ajuda a Fatima a descobrir uma partição válida das atrações (dados a , b e c), ou determina que não existe uma partição válida. Se existirem múltiplas partições válidas, podes descobrir e devolver qualquer uma delas.

Detalhes de implementação

Deves implementar a seguinte função:

```
int[] find_split(int n, int a, int b, int c, int[] p, int[] q)
```

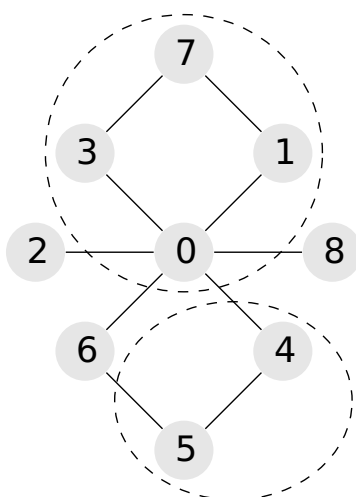
- n : o número de atrações.
- a , b e c : os tamanhos desejados para os conjuntos A , B e C .
- p e q : arrays de tamanho m , contendo as extremidades de cada estrada. Para cada i ($0 \leq i \leq m - 1$), $p[i]$ e $q[i]$ são as atrações ligadas pela estrada i .
- Esta função deve devolver um array de tamanho n . Seja este array denotado por s . Se não existir uma partição válida, s deve conter n zeros. Caso contrário, para $0 \leq i \leq n - 1$, $s[i]$ deve ser um de 1, 2, or 3 para indicar que à atração i é atribuído o conjunto A , B ou C , respetivamente.

Exemplos

Exemplo 1

Considera a seguinte chamada:

```
find_split(9, 4, 2, 3, [0, 0, 0, 0, 0, 0, 1, 3, 4, 5],  
           [1, 2, 3, 4, 6, 8, 7, 7, 5, 6])
```

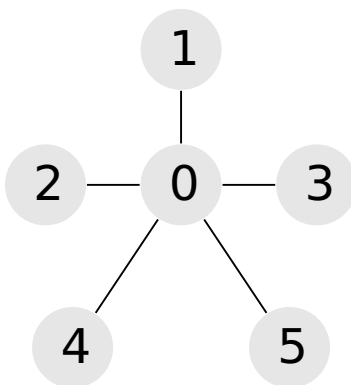


Uma possível solução correta é $[1, 1, 3, 1, 2, 2, 3, 1, 3]$. Esta solução descreve a seguinte partição: $A = \{0, 1, 3, 7\}$, $B = \{4, 5\}$, e $C = \{2, 6, 8\}$. Os conjuntos A e B são conexos.

Exemplo 2

Considera a seguinte chamada:

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



Nenhuma partição válida existe. Por isso mesmo, a única resposta correta é $[0, 0, 0, 0, 0, 0]$.

Restrições

- $3 \leq n \leq 100\,000$
- $2 \leq m \leq 200\,000$
- $1 \leq a, b, c \leq n$
- $a + b + c = n$
- Existe no máximo uma estrada entre cada par de atrações.
- É possível viajar entre qualquer par de atrações usando as estradas.
- $0 \leq p[i], q[i] \leq n - 1$ e $p[i] \neq q[i]$ para $0 \leq i \leq m - 1$

Subtarefas

1. (7 pontos) Cada atração está na extremidade de no máximo duas estradas.
2. (11 pontos) $a = 1$
3. (22 pontos) $m = n - 1$
4. (24 pontos) $n \leq 2500, m \leq 5000$
5. (36 pontos) Nenhuma restrição adicional.

Avaliador exemplo

O avaliador exemplo lê o input no seguinte formato:

- linha 1: $n\ m$
- linha 2: $a\ b\ c$
- linha $3 + i$ (para $0 \leq i \leq m - 1$): $p[i]\ q[i]$

O avaliador exemplo escreve uma única linha contendo o array devolvido por `find_split`.