Split the Attractions

Existem n atrações em Baku, numeradas de 0 a $n-1$. Também existem m ruas bidirecionais, numeradas de 0 à $m-1$. Cada rua conecta duas atrações diferentes. É possível viajar entre qualquer par de atrações usando as ruas.

Fatima está planejando visitar todas as atrações em três dias. Ela já deciciu que deseja visitar a atrações no primeiro dia, b atrações no segundo dia e c atrações no terceiro dia. Portanto, ela vai particionar as n atrações em três conjuntos A, B, e C cujos tamanhos são a, b, e c, respectivamente. Cada atração irá pertencer a exatamente um conjunto, então $a+b+c=n$.

Fatima deseja encontrar os conjuntos A, B, e C, de forma que **pelo menos dois** dos três conjuntos sejam **conexos**. Um conjunto S de atrações é dito conexo se é possível viajar entre qualquer par de atrações em S usando as ruas e sem passar por nenhuma atração que não esteja em S. Uma partição das atrações em conjuntos A, B, e C é dita **válida** se satisfaz as condições descritas acima.

Ajude Fatima a encontrar uma partição das atrações que seja válida (dados a, b, e c), ou determine que não existe nenhuma partição válida. Se houver mais de uma partição válida, você pode encontrar qualquer uma delas.

Detalhes de Implementação

Você deve implementar o seguinte procedimento:

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

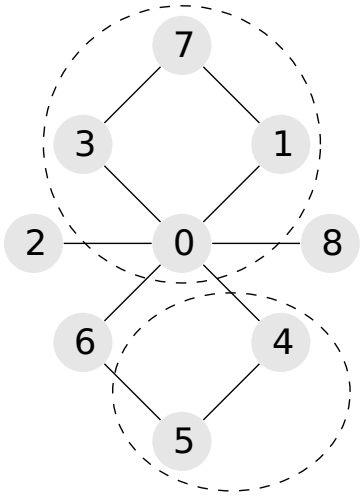
- n: o número de atrações.
- a, b, e c: os tamanhos desejados dos conjuntos A, B, e C.
- p e q: arrays de comprimento m, contendo as atrações de cada rua. Para cada i ($0 \leq i \leq m-1$), $p[i]$ e $q[i]$ são as duas atrações conectadas pela rua i.
- Esse procedimento deve retornar um array de comprimento n. Denote o array por s. Se não existir partição válida, s deve conter n zeros. Senão, para $0 \leq i \leq n-1$, $s[i]$ deve conter ou 1, ou 2, ou 3 para denotar que a atração i foi atribuída para o conjunto A, B, ou C, respectivamente.

Exemplos

Exemplo 1

Considere a seguinte chamada:

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

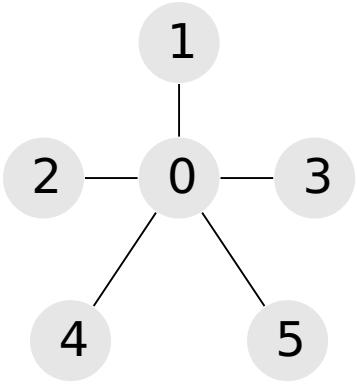


Uma possível solução correta é $[1, 1, 1, 1, 2, 2, 3, 3, 3]$. Essa solução descreve a seguinte partição: $A = 0, 1, 2, 3$, $B = 4, 5$, e $C = 6, 7, 8$. Os conjuntos A e B são conexos.

Exemplo 2

Considere a seguinte chamada:

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



Não existe nenhuma partição válida. Então, 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 rua entre cada par de atrações.
- É possível viajar entre qualquer par de atrações usando as ruas.
- $0 \leq p[i], q[i] \leq n - 1$ e $p[i] \neq q[i]$ para $0 \leq i \leq m - 1$

Subtarefas

1. (7 points) Cada atração está presente em no máximo duas ruas.
2. (11 points) $a = 1$
3. (22 points) $m = n - 1$
4. (24 points) $n \leq 2500, m \leq 5000$
5. (36 points) Nenhuma restrição adicional.

Corretor exemplo

O corretor exemplo lê a entrada 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 corretor exemplo imprime uma única linha contendo o array retornado pelo `find_split`.