

Тест по програмирање (3 година - МИГ)

1. Бипартитна пермутација

За даден позитивен цел број N , разгледајте ја секоја пермутација на сите броеви од 1 до N . Потребно е да се создадат две партиции, $P1$ и $P2$, од овие броеви така што $|\text{sum}(P1) - \text{sum}(P2)|$ е минимално, каде што сумата(X) го означува збирот на сите броеви во партицијата X .

Партицијата е дефинирана како непразно подмножество на пермутацијата. Со други зборови, пронајдете ја минималната апсолутна разлика помеѓу збирот на сите броеви во секоја партиција. Не можете да изоставите ниту еден број, секој број од 1 до N мора да биде дел од точно една партиција.

1.1. Пример:

Влез	Излез
5 // Број на тест случаи	Case 1: 1
2	Case 2: 0
3	Case 3: 0
4	Case 4: 1
5	Case 5: 1
6	

Да се реши со употреба на алчен (greedy) пристап.

2. Филозофски камења

Една од тајните одаи во Хогвортс е полна со филозофски камења.

Подот на одаите е покриен со $h \times w$ квадратни плочки, каде што има h редови плочки од напред (прв ред) кон назад (последниот ред) и w колони плочки од лево кон десно.

Секоја плочка има од 1 до 100 камења на неа. Хари Потер мора да зграпчи што е можно повеќе филозофски камења, со следниве ограничувања:

Тој започнува со избирање на која било плочка во првиот ред и ги собира филозофските камења на таа плочка. Потоа, тој се префрла на плочка во следниот ред, ги собира филозофските камења на плочката и така натаму додека не стигне до последниот ред.

Кога се движи од една плочка до плочка во следниот ред, може да се движи само до плочката веднаш под неа или дијагонално налево или надесно.

Со оглед на вредностите на h и w и бројот на филозофски камења на секоја плочка, напишете програма за да го пресметате максималниот можен број на филозофски камења што Хари може да ги зграби во едно патување од првиот ред до последниот ред.

Влез:

Првата линија се состои од еден цел број T , бројот на тест случаи. Потоа се задаваат бројот на плочки (редови и колони) и соодветно после тоа се задаваат бројот на камења на секоја

плочка.

Излез:

Излезот треба да се состои од T линии, ($1 \leq T \leq 100$), по една за секој тест случај. Секоја линија се состои од еден цел број, што е максималниот можен број на филозофски камења што Хари може да ги зграби, во едно патување од првиот до последниот ред за соодветниот тест случај.

Input:

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

Output:

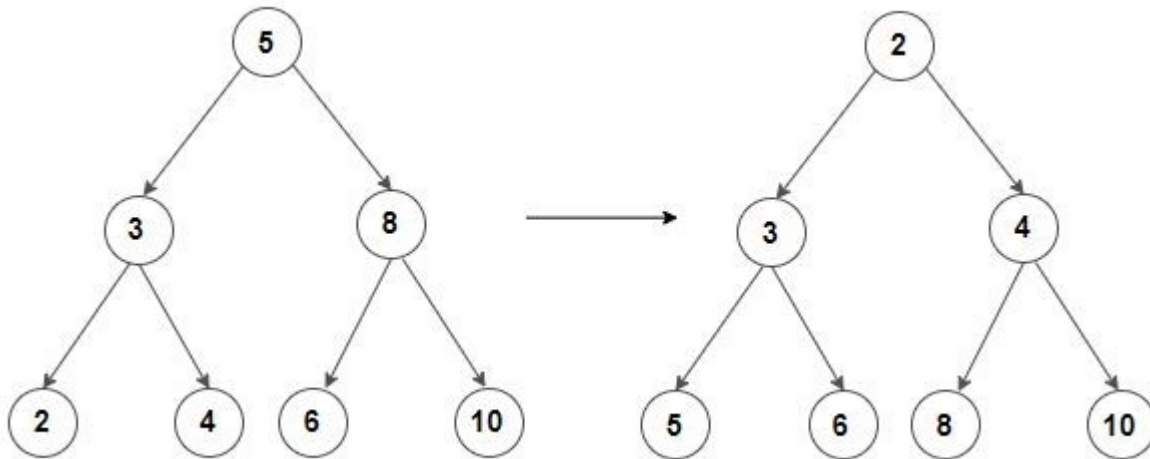
```
32
```

```
//7+1+8+5+4+7=32
```

2. Конвертирај бинарно пребарувачко дрво во min-heap дрво

За дадено комплетно бинарно пребарувачко дрво (BST), конвертирајте го во min-heap, односно конвертирајте го бинарното пребарувачко дрво во комплетно бинарно дрво во кое секој јазел има поголема вредност од неговиот родител и неговиот лев брат/сестра (ако го има).

Пример: Бинарното пребарувачко дрво дадено на левата слика ќе се конвертира во бинарното дрво во десната слика.



Почетниот код Ви е даден во z2.

3. Енклави

Дадена Ви е $m \times n$ бинарна матрица каде 0 репрезентира келија со вода додека 1 репрезентира келија со копно.

За чекор се смета движење од една копнена келија до друга соседна копнена келија (хоризонтално и вертикално) или излегување надвор од границите на матрицата.

Да се врати бројот од копнени келии во матрицата за кои не може да излеземе надвор од матрицата за било кој број на потези.

НАПОМЕНА: Решение со union-find ќе носи 100% од поените. Сите останати пристапи на решавање нема да носат повеќе од 70% од поените.

3.1. Пример 1:

0	0	0	0
1	0	1	0
0	1	1	0
0	0	0	0

Влез

```
grid = [ [0,0,0,0],  
         [1,0,1,0],  
         [0,1,1,0],  
         [0,0,0,0] ]
```

Излез: 3

Објаснување: Има 3 копнени келии (означени со 1) кои се ограничени со 0s, и една копнена келија која допира со работ матрицата.

3.2. Пример 2:

0	1	1	0
0	0	1	0
0	0	1	0
0	0	0	0

Влез:

```
grid = [[0,1,1,0],[0,0,1,0],[0,0,1,0],[0,0,0,0]]
```

Излез: 0

Објаснување: Сите келии со 1 се или на границата или можат да стигнат до границата.

Почетниот код Ви е даден во z3