

Problema 23 (HFY)

Echipa 2: Cabalau Razvan, Radu Catalin-Gabriel, Rotaru Ada
Grupa 212

I. Generarea cerintei.

```
1 from problem import Problem
2 import random
3 class Problem23(Problem):
4     def __init__(self):
5         # data = [8, 5, 3, 9, 2, 1, 7, 6]
6         # n = len(data)
7         n = random.randint(8, 12)
8         data = random.sample(range(12), n)
9         statement = f'Primiti vectorul: { str(data) }. \n'
10        statement += f'Faceti din acest vector un min-ansamblu folosind un numar minim de operatii si
11
12        super().__init__(statement, data)
```

Pentru a genera cerința cream vectorul de string-uri *statement* in care adăugam succesiv textul cerinței. Folosim functia *random.sample* pentru a genera sirul de numere. Sirul este de cel puțin 8 numere si de maxim 12 numere cu cifre cuprinse între 0 si 12.

II. Rezolvare .

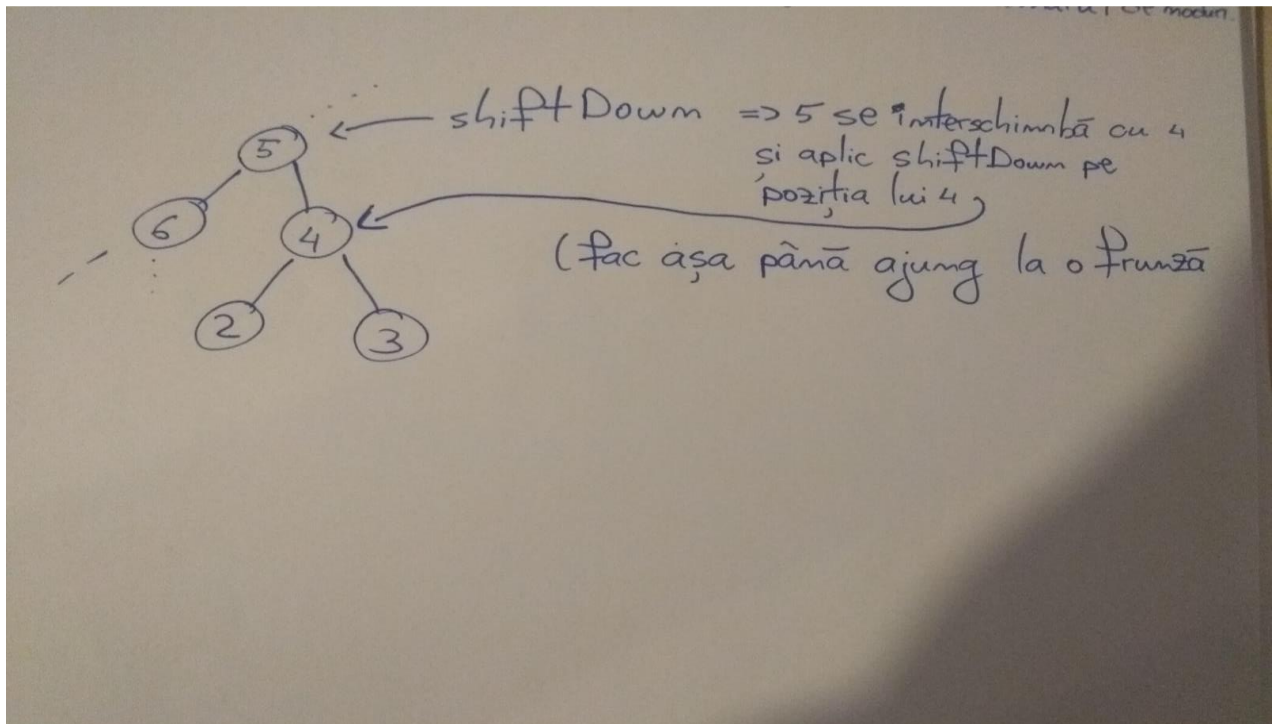
```
14 def shiftDown(self, H, n, index):
15     l = 2 * index + 1
16     r = 2 * index + 2
17     newIndex = index
18
19     if l < n and H[l] < H[newIndex]:
20         newIndex = l
21     if r < n and H[r] < H[newIndex]:
22         newIndex = r
23
24     H[index], H[newIndex] = H[newIndex], H[index]
25
26     if(newIndex != index):
27         self.shiftDown(H, n, newIndex)
28
```

In sirul de caractere *solution* adaugam ideea de rezolvare si modul prin care realizam solutia.

Aplicam *shiftDown* pentru nodurile de la $[n/2, 0]$ (in ordinea asta).

Shiftdown practic compara recursiv un nod *i* cu cei 2 copii si se interchimba cu cel mai mic, iar dupa aplicam shiftdown catre pozitia unde am interchimbat.

Dupa cum putem vedea in exemplul de mai jos:



Aici avem partea de afisare si de returnare a solutiei.

```
34 def solve(self):
35     v = self.data
36     solution = "Aplicam shiftDown pentru nodurile de la [n/2,0] (in ordinea asta).\n"
37     solution += "ShiftDown compara recursiv un nod i cu cei 2 copii si se interschimba cu cel mai mic
38     solution += "-> " + str(self.data[0]) + " se interschimba cu " + str(self.data[1]) + " "
39     solution += "si aplic algoritmul pana ajung la o frunza.\n"
40
41     n = len(self.data)
42     for i in range(int(n / 2), -1, -1):
43         self.shiftDown(self.data, n, i)
44     #return self.data
45     solution += "In final vom obtine min-ansamblul: \n"
46     solution += "" + str(self.data) + ""
47     return solution
48
49
50 # todo: delete this
51 obj = Problem23()
52 print(obj.statement)
53 print(obj.solve())
```