

TEMA SEMINAR 1
(Algoritmi fundamentali)

1. Problema cotidiana rezolvata cu ajutorul garfurilor.

Enuntul problemei:

Un turist a ajuns in Bucuresti si doreste sa viziteze cat mai multe atractii turistice. Pentru fiecare zona a Bucurestiului se cunoaste numarul de atractii turistice. Singurul mijloc de transport cu care se poate deplasa turistul este metroul, existand magistrale intre diverse zone. Magistralele sunt inregistrate in aplicatia folosita de turist in ordinea cronologica a constructiei lor. Turistul si-a stabilit traseul astfel: pornind dintr-o zona a orasului, ia metroul spre urmatoarea zona si asa mai departe pana cand nu mai are mijloc de transport. In cazul in care are mai multe posibilitati de a lua metroul, acesta va alege magistrala care il va duce intr-o zona cu cat mai multe atractii turistice sau magistrala cea mai veche, in caz de egalitate.

- Din ce zona a Bucurestiului ar trebui sa plece turistul pentru a vizita cat mai multe atractii turistice?
- Cate atractii turistice viziteaza turistul?

Ideea algoritmului:

Salvam intr-o lista numarul de atractii al fiecarei zone si intr-un dictionar graful asociat retelei de metrou (liste de adiacenta).

Pornind din fiecare oras, salvam traseul generat de proprietatile calatoriei (zona cu cat mai multe atractii turistice sau magistrala cea mai veche). La final, facem suma atractiilor turistice si o alegem pe cea mai mare, returnand orasul din care am plecat.

Calatoria are la baza un BFS. Retinem la fiecare pas zona vizitata curent pentru a nu trece de doua ori prin ea. Uitandu-ne la numarul de atractii al zonelor vecine, alegem zona cu cel mai mare numar de atractii sau pe prima din punct de vedere lexicografic. Continuam calatoria pe acea zona daca nu a fost vizitata deja. In cazul in care mai putem vizita alte zone, mergem mai departe, altfel am gasit traseul ce pleaca dintr-un nod al grafului.

Complexitatea parcurgerii este $O(n+m)$. Cum facem o parcurgere pentru fiecare nod, complexitatea totala va fi $O(n(n+m))$.

```
f = open("metrou.in")

numar_zone = int(f.readline())

numar_atractii_pe_zone = [-99999]
aux = [int(x) for x in f.readline().split()]
for elem in aux:
    numar_atractii_pe_zone.append(elem)
```

```

harta = {}
for i in range(1, numar_zone+1):
    harta[i] = []
rand = f.readline()
while rand != "":
    if rand != "":
        xy = [int(x) for x in rand.split()]
        x = xy[0]
        y = xy[1]
        harta[x].append(y)
        harta[y].append(x)
    rand = f.readline()

zone_vizitate = [0 for i in range(numar_zone+1)]
posibilitati = []
drum = []

def calatorie(zone):
    salvam_zona = 0
    maxim = 0
    zone_vizitate[zone[-1]] = 1
    for zona_vecina in harta[zone[-1]]:
        if zone_vizitate[zona_vecina] == 0 and
numar_attractii_pe_zone[zona_vecina] > maxim:
            salvam_zona = zona_vecina
            maxim = numar_attractii_pe_zone[zona_vecina]
    zone.append(salvam_zona)
    zone_viitoare = harta[salvam_zona]
    mai_departe = 0
    for x in zone_viitoare:
        mai_departe += zone_vizitate[x]
    if mai_departe != len(zone_viitoare):
        calatorie(zone)
    else:
        posibilitati.append(zone)
    return

for i in range(1, numar_zone+1):
    calatorie([i])
    zone_vizitate = [0 for i in range(numar_zone + 1)]

sume = [-99999]
for pos in posibilitati:
    suma = 0
    for elem in pos:
        suma += numar_attractii_pe_zone[elem]
    sume.append(suma)

for i in range(numar_zone):
    print("Daca turistul pleaca din zona ", i+1, ", acestea sunt zonele vizitate: ", posibilitati[i])

```

```
print("Pentru a vizita un numar maxim de atractii turistice, turistul trebuie  
sa porneasca din zona ",  
      sume.index(max(sume)), ", vizitand in total ", max(sume), " de atractii  
turistice.")
```

Exemplu:

```
11  
3 1 2 3 2 1 1 1 3 4 10  
1 2  
1 3  
2 4  
2 5  
2 6  
3 8  
3 7  
8 11  
7 9  
7 10
```

Rezultat:

```
Daca turistul pleaca din zona 1 , acestea sunt zonele vizitate: [1, 3, 8,  
11]  
Daca turistul pleaca din zona 2 , acestea sunt zonele vizitate: [2, 1, 3,  
8, 11]  
Daca turistul pleaca din zona 3 , acestea sunt zonele vizitate: [3, 1, 2,  
4]  
Daca turistul pleaca din zona 4 , acestea sunt zonele vizitate: [4, 2, 1,  
3, 8, 11]  
Daca turistul pleaca din zona 5 , acestea sunt zonele vizitate: [5, 2, 1,  
3, 8, 11]  
Daca turistul pleaca din zona 6 , acestea sunt zonele vizitate: [6, 2, 1,  
3, 8, 11]  
Daca turistul pleaca din zona 7 , acestea sunt zonele vizitate: [7, 10]  
Daca turistul pleaca din zona 8 , acestea sunt zonele vizitate: [8, 11]  
Daca turistul pleaca din zona 9 , acestea sunt zonele vizitate: [9, 7, 10]  
Daca turistul pleaca din zona 10 , acestea sunt zonele vizitate: [10, 7, 9]  
Daca turistul pleaca din zona 11 , acestea sunt zonele vizitate: [11, 8, 3,  
1, 2, 4]  
Pentru a vizita un numar maxim de atractii turistice, turistul trebuie sa  
porneasca din zona 4 , vizitand in total 20 de atractii turistice.
```