Laborator 7

Sîrbu Matei-Dan

26 noiembrie 2020

Exercițiul 1

Breviar teoretic

Fie G un graf orientat. G este un *arbore cu rădăcina* r, dacă există G un vârf r din care oricare alt vârf poate fi ajuns printr-un drum unic.

Definiția este valabilă și pentru cazul unui graf neorientat, alegerea unei rădăcini fiind însă în acest caz arbitrară: orice arbore este un arbore cu rădăcină, iar rădăcina poate fi fixată în oricare vârf al său. Aceasta, deoarece dintr-un vârf oarecare se poate ajunge în orice alt vârf printr-un drum unic.

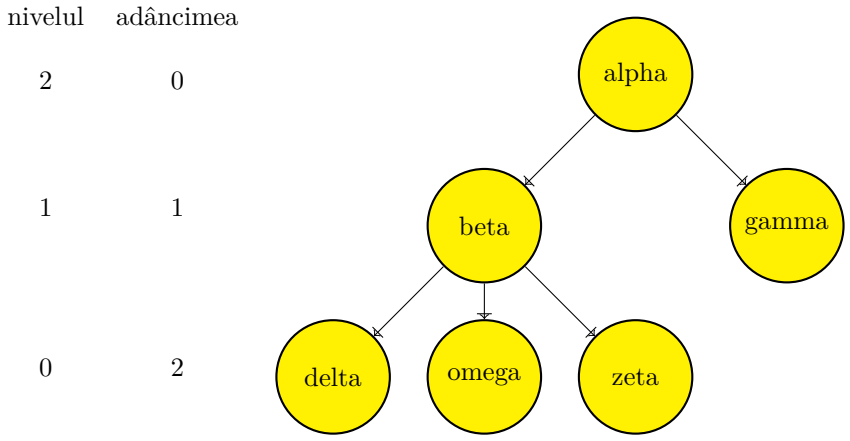

Figura 1: Un arbore cu rădăcină.

Când nu va fi pericol de confuzie, vom folosi termenul „arbore”, în loc de termenul corect „arbore cu rădăcină”. Cel mai intuitiv este să reprezentăm un arbore cu rădăcină, ca pe un arbore propriu-zis. În figura 1, vom spune că *beta* este *tatăl* lui *delta* și fiul lui *alpha*, că *beta* și *gamma* sunt *frați*, că *delta* este un *descendent* al lui *alpha*, iar *alpha* este un *ascendent* al lui *delta*.

Exercițiul 2

Presupunem, pentru simplificare, că vârfurile sunt numerotate, $V = \{1, 2, ..., n\}$, vârful 1 fiind sursa, și că matricea L dă lungimea fiecărei muchii, cu $L[i, j] = +\infty$, dacă muchia (i, j) nu există. Soluția se va construi în tabloul $D[2..n]$. Algoritmul este:

```
procedure DIJKSTRA(L[1..n, 1..n])                    ▷ inițializare
    C ← {2, 3, ..., n}                    ▷ S = V \C există doar implicit
    for i ← 2 to n do
        D[i] ← L[1, i]                                  ▷ bucla Greedy
    end for
    repeat n − 2 times
        v ← vârful din C care minimizează D[v]
        C ← C\{v}                          ▷ și, implicit, S ← S ∪ {v}
        for fiecare w ∈ C do
            D[w] ← min(D[w], D[v] + L[v, w])
        end for
    return D
end procedure
```

Pentru graful din Figura 2, pașii algoritmului sunt prezentați în Tabelul 1.

Observăm că D nu se schimbă dacă mai efectuăm o iterație pentru a-l scoate și pe $\{2\}$ din C. De aceea, bucla greedy se repetă de doar $n - 2$ ori.

Se poate demonstra următoarea proprietate:

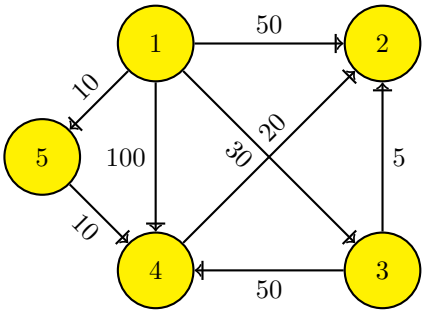

Figura 2: Un graf orientat.

Pasul	v	C	D
inițializare	—	{2, 3, 4, 5}	[50, 30, 100, 10]
1	5	{2, 3, 4}	[50, 30, 20, 10]
2	4	{2, 3}	[40, 30, 20, 10]
3	3	{2}	[35, 30, 20, 10]

Tabelul 1: Algoritmul lui Dijkstra aplicat grafului din Figura 2.