

## Radare – Descrierea solutiei

*Stud. Andrei Pârvu, Univeristatea POLITEHNICA București*  
*Stud. Bogdan Tătăroiu, Univeristatea POLITEHNICA București*

### Solutia 1 – 30 puncte - $O(2^N * N)$

Se vor genera toate submultimile de muchii, o muchie selectata reprezentand ca exista un radar pe acea muchie. Cu ajutorul unei parcurgeri (in adancime sau latime) se vor determina nodurile conectate de nodul 1, si se vor numara toate variantele care au suma P.

### Solutia 2 – 50 puncte - $O(N * P^2)$

Vom construi o dinamica  $A[i][j]$  – numarul de posibilitati de a selecta muchii din subarborele lui i, astfel incat din radacina subarborelui sa se poata ajunge in exact j noduri. Aceasta solutie se aplica pentru testele in care timpii de vizitare a tuturor oraselor sunt 1.

Pentru a calcula valoarea dinamicii in subarborele i, se calculeaza mai intai valorile pentru fiecare din fiii acestuia. Vom construi inca o dinamica  $B[a][b]$  – numarul de posibilitati de a selecta muchii din subarborii primilor a fii ai lui i astfel incat sa se poata ajunge in exact b noduri din radacina. Modul de calculare a acestei dinamici este foarte asemanator cu problema Rucsacului (Knapsack), avand 2 cazuri pentru fiecare fiu: fie taiem muchia dintre i si fiu (caz in care muchiile din subarborele fiului pot avea orice configuratie), fie variem numarul de noduri c din subarborele fiului si ne folosim de  $A[fiu][c]$ .

### Solutia 3 – 60 puncte - $O(N * P^2)$

Pentru 60 de puncte, solutia se construiesc in mod similar cu solutia de 50 de puncte.  $A[i][j]$  – numarul de posibilitati de a selecta muchii din subarborele lui i, astfel incat din radacina subarborelui sa se poata ajunge in noduri avand suma totala a timpilor de vizitare exact j. Implementarea este asemenatoare cu cea de 50 de puncte.

### Solutia 4 – 100 puncte - $O(N * P)$

Solutia de 100 de puncte porneste de la urmatoarea observatie: daca pe o muchie (x, y) este un radar (x fiind mai apropiat de radacina decat y) atunci muchiile din subarborele lui y pot avea orice configuratie. Astfel ne vine ideea de a liniariza arborele si de a face urmatoarea dinamica:  $A[i][j]$  – numarul de moduri de a ajunge pe pozitia i a liniarizarii, suma nodurilor accesibile din 1 fiind j. In acest moment avem doua variante: fie taiem muchia parinte a nodului cu timpul de intrare i, fie adaugam acest nod la configuratia curenta. Daca muchia este taiata, putem sari intregul subarbor al nodului, ajungand pe pozitia  $timp\_sfarsit[nod]$ . Astfel:

- $A[i + 1][j + valoare[nod\_timp\_i]] += A[i][j]$ , daca nu taiem muchia
- $A[timp\_sfarsit[nod\_timp\_i]][j] += A[i][j]$ , daca taiem muchia