Tabăra de pregătire a Lotului Naţional de Informatică
Focşani, 15-22 mai, 2010
Baraj 1 Seniori

Problema Regat Autor Airinei Adrian
Descrierea soluţiei

Fixăm rădăcina arborelui în nodul `1`. Se face o parcurgere dfs din nodul `1` şi se reţine pentru fiecare nod timpul la care a fost atins în parcurgerea dfs.
Astfel, toate nodurile din subarborele cu rădăcina în `x` reprezintă un interval continuu. Iniţial calculăm distanţele de la nodul `1` la celelalte noduri şi le introducem într-un arbore de intervale. Observăm că atunci când coborâm o muchie în arbore sau cand urcăm o muchie în arbore distanţele se modifică în felul următor: să zicem că suntem în nodul `x` şi coborâm pe arbore în nodul `y` (fiu al lui `x`), atunci dacă subarborele cu rădăcina în `y` are asociat intervalul `[a,b]`, nodurile din intervalul `[1,a-1]` şi `[b+1,a]` cresc distanţa de la nodul `x` la ele cu lungimea muchiei de la `x` la `y`, iar cele din intervalul `[a,b]` scad cu această lungime.
Când urcăm în arbore va fi invers, `[1,a-1]` şi `[b+1,a]` vor creşte şi `[a,b]` va scădea. Cand suntem intr-un nod `x` mai trebuie sa gasim cele `K[x]` valori maxime din arborele de intervale. Vom scoate pe rând elementul maxim din arbore şi vom avea grijă ca apoi să introducem din nou cele `K[x]` valori maxime pe care le-am scos.
Altfel spus, am redus problema noastră la urmatoarea problemă în care putem folosi arborii de intervale:
- adună o valoare pe un interval
- selectează valoarea maximă din arbore

Cum `K[1]+K[2]+...+K[N]` este de ordinul `N` şi noi facem maxim `K[1]+K[2]+...+K[N]` operaţii, complexitatea va fi `O(N*logN)` ca timp şi `O(N)` ca memorie.