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

Problema KDtree, Autor Cosmin Gheorghe
Descrierea soluţiei

Pentru 70% din puncte problema se poate rezolva cu programare dinamică. Agăţăm arborele într-un nod oarecare. Vom calcula `D[i][j]` = numărul minim de muchii ce trebuiesc tăiate în subarborele `i`, astfel încât toţi subarborii desprinşi să fie KDtrees iar subarborele rămas ce conţine nodul `i` să aibe adâncimea `j`. Adâncimea unui arbore este distanţa maximă de la rădăcina la oricare dintre frunze (cele nou formate după desprinderea unor subarbori). Pentru a calcula `D[i][j]` vom proceda în felul următor. Vom alege pe rând câte un fiu `fx` al lui `i` şi vom avea grijă ca subarborele determinat de acest fiu (în care am tăiat muchiile necesare) să aibe înalţimea `j - 1` (el va asigura înaltimea subarborelui `i`, adică `j`). Toţi ceilalţi fii `fk` ai lui `i` trebuie să aibă fiecare câte un subarbore (cei ramaşi după ce am tăiat muchiile necesare) a cărui inaltime `hk` să respecte următoarele două condiţii: `hk + j + 1 <= K` (adică drumul maxim ce trece prin subarborii `fx` si `fk` să aibe lungimea maxim `K`) şi `hk <= j - 1` (nu vrem să depaşim înalţimea `j` a subarborelui `i`). Aşadar recurenţa pentru `D[i][j]` se conturează astfel:

$$D[i][j] = Min\left\{ D[f_x][j-1] + \sum_{\substack{f_k \text{ fiu al lui i,} \\ f_k \neq f_x}} \left(\text{Min}\left\{ D[f_k][h_k] \mid h_k \le \min(K-j-1, j-1) \right\} \right) \;\middle|\; f_x \text{ fiu al lui } i \right\}$$

Această dinamică se poate implementa uşor în `N * K^3`, apoi se poate reduce la `N*K^2` pentru `40` de puncte şi în final la `N * K` pentru `70` de puncte.

Pentru `100` de puncte vom aplica o strategie greedy care se poate deduce din dinamica explicată mai sus. Vom parcurge nodurile într-o maniera bottom-up de la frunze spre rădăcină. Pentru fiecare nod curent calculăm `Hi`, inălţimea subarborelui `i`. Dacă pentru toţi fii `fi` ai lui `i`, subarborii determinaţi de ei sunt KDtrees putem verifica şi dacă subarborele `i` este tot `KDtree` considerând doar drumurile ce îl conţin pe `i`. Astfel, fie `h1` si `h2` înalţimile celor mai înalţi doi subarbori ai lui `i`. Dacă `h1 + h2 + 2` (drumul constând din aceste două inalţimi si nodul `i`) este mai mare decat `K` atunci subarborele `i` nu este un KDtree. Este optim să “desprindem” cel mai înalt subarbore al lui `i`. Adică daca `fi` are înalţimea maximă vom tăia muchia `i-fi`. Putem forma astfel urmatoarea strategie: cât timp drumul prin cei mai inalţi doi subarbori şi nodul `i` este mai mare decat `K` desprindem cel mai înalt subarbore. Trebuie să avem grijă să actualizăm înaltimea `Hi` a nodului `i`, în funcţie de subarborii desprinşi. La sfârşitul parcurgerii toţi arborii rămaşi după “desprinderile” efectuate vor fi KDtrees. Această soluţie se poate implementa în timp `O(N log N)` si obţine `100` de puncte.