Tcast – descrierea solutiei

Mugurel-Ionuţ Andreica, Politehnica Bucureşti

Se va stabili radacina arborelui in nodul 1, definindu-se astfel relatii de tip tata-fiu intre nodurile vecine. Se vor calcula urmatoarele valori:

- **Tmin[i,t]**=timpul minim dupa care toate nodurile din subarborele nodului i primesc mesajul, daca nodul i primeste mesajul la momentul t ($1 \le i \le N$, $0 \le t \le T+N$).

Pentru o frunza i, avem Tmin[i,t]=t. Fie i un nod intern al arborelui, care are K fii: f_1, f_2, …, f_K. Pentru fiecare moment de timp t, vom determina primele K momente de timp (mai mari sau egale cu t) la care nodul i nu este verificat. Fie aceste momente de timp $t_1<t_2<\ldots<t_K$. In fiecare dintre aceste momente de timp, nodul i va trimite mesajul unuia dintre cei K fii. Fiecarui fiu f_j i se va atribui cate un numar distinct m(j) ($1 \le m(j) \le K$), avand semnificatia ca nodul i trimite mesajul fiului f_j la momentul $t_{m(j)}$. Trebuie sa determinam o atribuire a valorilor m(j), astfel incat **max{Tmin[f_1,1+$t_{m(1)}$], T_{min}[f_2,1+$t_{m(2)}$], …, Tmin[f_K,1+$t_{m(K)}$]}** sa fie **minim**. Pentru a rezolva eficient aceasta problema, ne vom folosi de urmatoarea proprietate: **Tmin[x,t]≤Tmin[x,t+1]**.

<u>Solutie O(N·log(N)·T·log(T))</u>

Vom cauta binar valoarea Tmin[i,t]. Sa presupunem ca valoarea testata in cadrul cautarii binare este T_{test}. Pentru fiecare fiu f_j, vom determina cea mai mare valoarea lmom(j), astfel incat Tmin[f_j, 1+$t_{lmom(j)}$]≤T_{test}. Daca nu exista nici o astfel de valoare, vom considera lmom(j)=0. Pentru a determina lmom(j) vom realiza o cautare binara. Vom sorta apoi fiii in ordine crescatoare a valorilor lmom(j). Daca exista un fiu f_j aflat pe pozitia p in ordinea sortata si avem lmom(j)<p, atunci timpul ales T_{test} este prea mic si trebuie incercata o valoare mai mare. Pentru a determina eficient momentele de timp t_1, t_2, …, t_K, vom observa ca atunci cand trecem de la un moment t la momentul t+1, momentele de timp t_1, …, t_K ori raman aceleasi, ori se renunta la momentul t_1 si se adauga urmatorul moment de timp mai mare decat t_K la care nodul i nu este verificat. Aceasta solutie are complexitatea **O(N·T·log(T)· log(N))**.

Solutie O(N·log(N)·T)

Aceasta solutie este asemanatoare cu cea anterioara, insa vom inlocui cautarea binara a momentului de timp cu o cautare scventiala. Pentru un nod i, vom calcula, in ordine, Tmin[i,0], Tmin[i,1],…, Tmin[i,T+N]. Pentru a calcula Tmin[i,t] (t>0) vom cauta secvential rezultatul, incepand cu momentul Tmin[i,t-1] (folosindu-ne de proprietatea mentionata anterior). In felul acesta, vom realiza O(T) teste pentru toate momentele de timp t.

Solutie O(N·T)

Solutia poate fi optimizata pana la complexitatea **O(N·T)**. Pentru aceasta, trebuie sa determinam mai eficient valorile lmom(j) si sa sortam mai repede fiii unui nod in ordine crescatoare a valorilor lmom. Cand trecem de la un moment t la momentul urmator t+1, este clar ca $t_{lmom(j)}$ la momentul t este mai mic sau egal decat $t_{lmom(j)}$ la momentul t+1. In felul acesta, putem folosi aceeasi idee de cautare secventiala folosita pentru a inlocui cautarea binara dupa timp. Pentru a realiza sortarea fiiilor in timp liniar, vom folosi un algoritm asemanator cu countsort.

Ca o optimizare, pentru momentele de timp t>T, avem Tmin[i,t]=1+Tmin[i,t-1].