

Problema 3 – venus

100 de puncte

Descrierea unor soluții posibile

prof. Emanuela Cerchez, Colegiul Național "Emil Racoviță" Iași

Soluția 1

prof. Emanuela Cerchez, Colegiul Național "Emil Racoviță" Iași

În primul rând observăm că dacă o comandă nu poate fi onorată la timp, putem să o lăsăm la final, pentru oricum nu vom primi valoarea ei. Deci comenzile neonorate la timp vor fi toate plasate la sfârșitul planificării optimale, nu contează în ce ordine.

Pierderea va fi suma valorilor acestor comenzi.

Ca urmare ne interesează în mod special lista comenzilor care pot fi onorate la timp (să o numim pe scurt

Lista)

Observăm de asemenea că este convenabil să onorăm mai întâi comenzile cele mai valoroase, deci vom ordona comenzile descrescător după valoare.

Vom parcurge comenzile în ordinea descrescătoare a valorii lor și pentru fiecare comandă analizăm dacă o putem sau nu onora la timp. Pentru aceasta vom căuta o poziție în *Lista* în care comanda respectivă ar putea fi inserată (fără a afecta alte comenzi, adică livrarea comenzilor existente deja în *Lista* trebuie să rămână la timp, deoarece aceste comenzi sunt mai valoroase).

Dacă o astfel de poziție nu este găsită vom considera că această comandă este neonorată.

Pentru a putea verifica dacă inserarea este posibilă, vom reține comenzile în *Lista* în ordinea crescătoare a timpului lor de predare. Parcurgem comenzile secvențial în ordine inversă și căutăm poziția în care putem insera comanda curentă, astfel încât *Lista* să rămână ordonată după timpul de predare, iar comenzile parcurse pot fi amânate cu timpul de predare al comenzii curente.

Deoarece este complicat să lucrăm cu date și cu ore, chiar de la citire vom calcula timpul de predare în ore.

Soluție alternativă

Prof. Florentina Ungureanu, Colegiul Național de Informatică, Piatra Neamț

Se declară un vector p cu maximum $366 \cdot 24$ elemente, pozițiile 1, 2, ..., $366 \cdot 24/t$ corespunzând perioadelor $t, 2 \cdot t, 3 \cdot t$ ș.a.m.d.

Se transformă timpii de finalizare ai comenzilor în ore și se reține pentru fiecare comandă într-un câmp nt numărul întreg de perioade t după care aceasta trebuie finalizată.

Se sortează descrescător după valoare comenzile.

Se parcurge șirul obținut și pentru fiecare comandă $v[i]$:

- se verifică dacă $p[v[i].nt]$ este 0, caz în care aceasta se planifică executarea comenzii în perioada $v[i].nt$, iar $p[v[i].nt] = v[i].nr$ (nr. inițial al comenzii);

- dacă $p[v[i].nt] \neq 0$, se caută poziția k ($k \in \{v[i].nt-1, v[i].nt-2, \dots, 1\}$) cu proprietatea $p[k]=0$, caz în care se planifică executarea comenzii în perioada k , se bifează într-un vector boolean comenzile executate. În cazul în care nu se găsește o poziție k conform specificațiilor de mai sus, comanda rămâne restantă;

- valoarea totală a comenzilor restante se determină scăzând din valoarea totală a comenzilor, valoarea fiecărei comenzi planificate;

- se afișează comenzile planificate parcurgând șirul p construit, iar cele restante în ordine crescătoare a numărului inițial (ordinea executării acestora nefiind importantă).