

Tehnici Avansate de Programare

Laborator Greedy

Varianta 2

Problema 3 : Maximizarea profitului cu respectarea termenelor limită.

Soluție:

Se sortează toate activitățile descrescător după termenul limită. Timpul curent îl setăm să fie egal cu cel mai mare timp limită. La fiecare pas se adaugă toate activitățile cu timpul limită mai mare sau egal ca timpul curent. Se selectează capul heapului la fiecare pas.

Corectitudine.

Soluția este o submulțime de activități. Fie G soluția greedy și

O - soluția optimă maxim asemănătoare cu soluția greedy.

$$G = \{g_1, g_2, \dots, g_k\} \quad O = \{o_1, o_2, \dots, o_k\}$$

Dacă $O \neq G \Rightarrow \exists i \text{ aș. } g_i \neq o_i$

a) $g_i \notin O$.

Construim $O' = \{o_1, o_2, \dots, g_i, o_{i+1}, \dots, o_k\}$

pentru că g_i este activitatea cu profit maxim care poate fi realizată la pasul i , O' este soluție mai bună decât O \square

b) $g_i \in O$ dar la alt pas.

Construim $O' = \{o_1, o_2, \dots, g_i, o_{i+1}, \dots, o_k\}$

$$O = \{o_1, o_2, \dots, o_i, \dots, g_i, \dots, o_k\}$$

O' este soluție optimă mai asemănătoare cu G decât O \square

Complexitate

sortare $O(n \log n)$

inserare în heap $O(\log n)$

ștergere de pe heap $O(\log n)$

afișare $O(n)$

citire $O(n)$

$O(n \log n)$

} de maxim $2n$ ori