The rod cutting problem

Mănescu Marius-Alexandru

June 5, 2016

Anul: I Grupa: 10104 A Specializarea: Calculatoare română

Abstract

Documentul de faţă descrie problema abordată ş modul în care acesta poate fi rezolvată, utilizând diverse tehnici de programare

1 Introducere

În cele ce urmează voi încerca să descriu cât mai bine proiectul ales, modul în care se comportă şi care este scopul acestuia.

Codul este disponibil la acestă adresă https://github.com/Manescu/proiect_prg/blob/master/the%20rod%20cutting%20problem/rod_cutting.c

2 Descrierea proiectului

Problema presupune împarţirea unei tije în n elemente nu neapărat egale, fiecare element având un preţ diferit.In felul acesta trebuie să descoperim cea mai bună metodă de a decupa bara astfel încât să obţinem cel mai bun profit.

1. Pentru început trebuie să generăm dimensiunea tijei şi tebelul de preţuri astfel:

   $n \leftarrow$ random $1 - 100$

   afisam n

2. Afişăm tabelul de preţuri

   Deoarece dimensiunea tijei poate varia într-un interval destul de mare, vectorul (tabelul) trebuie alocat dinamic, deasemenea valorile componentelor tijei, pe care le salvăm in vector, trebuiesc scrise automat prin functia **random**.

   Structura de vector/matrice are avantajul simplităţii si economiei. Alocarea dinamica este o soluţie mai flexibilă,care foloseşte mai bine memoria şi nu impune limitări arbitrere asupra utilizări unor programe.

   - Operatorul **sizeof** este utilizat pentru a determina numărul de octeţi necesari unui tip de variabile.

   $x \leftarrow (int*)calloc(n, sizeof(int))$

   $srand((unsigned)time(NULL))$

   pentru $i \leftarrow 1$ la n **executa**

   $x_i \leftarrow random\ 1 - 1000$

3. Generăm o matrice de $n - 1$ linii si n coloane, pe care o alocam dinamic, in care salvăm calculele în scopul obţinerii soluţiei optime:

   real $**a$

$a \leftarrow (real**)malloc(n * sizeof(real*))$

pentru $i \leftarrow 1$ *lan* **executa**

$a_i \leftarrow calloc(n, sizeof(real))$

pentru $i \leftarrow 1$ *la* $n-1$ **execută**

pentru $j \leftarrow 1$ *la* n **execută**

daca $j >= i$ **atunci**

$a_{ij} \leftarrow max(a_{i-1,j}, x_i + a_{i,j-i})$

altfel

$a_{i,j} = a_{i-1,j}$

3 Exemple:

1. Pentru $n = 5$ (dimensiunea tijei) avem:

1	2	3	4
2 $	5 $	7 $	8 $

▷ Pentru o tăietura de dimensiune 1 preţul este de 2 $

▷ Pentru o tăietura de dimensiune 2 preţul este de 5 $

▷ Pentru o tăietura de dimensiune 3 preţul este de 7 $

▷ Pentru o tăietura de dimensiune 4 preţul este de 8 $

Soluţia optimă pentru aceste date de intrare este: **1,2,2** sau **2,3**

2. Pentru $n = 9$ avem:

1	2	3	4	5	6	7	8
3 $	5 $	8 $	9 $	10 $	17 $	17 $	20 $

▷ Pentru o tăietura de dimensiune 1 preţul este de 3 $

▷ Pentru o tăietura de dimensiune 2 preţul este de 5 $

▷ Pentru o tăietura de dimensiune 3 preţul este de 8 $

▷ Pentru o tăietura de dimensiune 4 preţul este de 9 $

▷ Pentru o tăietura de dimensiune 5 preţul este de 10 $

▷ Pentru o tăietura de dimensiune 6 preţul este de 17 $

▷ Pentru o tăietura de dimensiune 7 preţul este de 17 $

▷ Pentru o tăietura de dimensiune 8 preţul este de 20 $

Soluţia optimă pentru aceste date de intrare este: **6,3** cu un profit de 25 $

3.1 Listă funcţii

Mai jos este prezentă o listă de fucţii care sunt utilizate in program:

- $int\ max(int\ a, int\ b)$ ce returnează maximul intre doua numere.
  - ▷ a şi b sunt parametri de tip intreg
- $float\ rod\_cutting()$,reprezintă esenţa programului, unde:
  - ▷ n, i, j sunt parametri de tip intreg
  - ▷ $**a$ - declararea matricei a ce urmează a fi aloctă dinamic
  - ▷ $*x$ - un vector ce urmează a fi alocat dinamic
  - ▷ ar trebui să returneze soluţia optimă pentru problemă

References

[1] Brian W. Kernighan and Dennis Ritchie The C Programming Language (2nd Edition)

[2] Peter van der Linden Expert C Programming

[3] https://www.sharelatex.com/learn/, accesat în data de 5 mai 2016