**⓫** **Metoda Greedy**

***OBIECTIVELE LABORATORULUI***



* Aprofundarea metodei Greedy;
* Aplicabiliatea corectă a noțiunilor prezentate prin rezolvarea unor probleme asemănătoare cu cele exemplificate în cadrul laboratorului.

1. **Aspecte teoretice**

Metoda Greedy se folosește în general în situația în care este dată o mulțime A și se cere să se găsească o submulțime B a sa care să îndeplinească anumite condiții sau un anumit criteriu de optim. Această metodă nu își propune găsirea celor mai bune soluții ale problemei date ci doar a uneia dintre ele care îndeplinește criteriul de optimizare ales.

Mecanismul general al metodei Greedy este următorul:

1. Se inițializează mulțimea soluțiilor cu mulțimea vidă (B=)
2. Se alege un element x din mulțimea A
3. Se verifică dacă un element ales poate fi adăugat la mulțimea soluțiilor, dacă da atunci va fi adăugat: B=B
4. Se continuă repeditiv cu pasul II până când au fost determinate toate elementele din mulțimea soluțiilor.

Există două variante de proceduri pentru metoda Greedy:

Metoda 1:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | **procedura** Greddy\_1 **(**A**,** n**,** B**)** **este**  B**=**  **pentru** i**=**1**,**n **executa**  x **=** ALEGE**(**A**,**n**,**i**)**  **daca** POSIBIL**(**B**,**x**)** **atunci**  ADAUG**(**B**,**x**)**  @  @  **sfarsit** |

Metoda 2:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | **procedura** Greddy\_2 **(**A**,** n**,** B**)** **este**  PREL(A,n)  B**=**  **pentru** i**=**1**,**n **executa**  **daca** POSIBIL**(**B**,**a1**)** **atunci**  ADAUG**(**B**,**a1**)**  @  @  **sfarsit** |

**Exercițiu 1: Problema continuă a rucsacului**



Fie un rucsac de volum V și n obiecte având fiecare volumele vi și prețul pi. Se cere să se umple rucsacul astfel încât prețul total al obiectelor introduse să fie maxim.

Se va considera că se poate introduce și o parte dintr-un obiect care nu încape în rucsac, până la ocuparea întregului volum al rucsacului.

***Rezolvarea prin metoda Greedy:***

Se introduc în rucsac obiectele unul câte unul, în ordinea descrescătoare a raportului pi/vi („densitate valorică”), astfel încât suma volumelor introduse să fie mai mică sau cel mult egală cu V.

Pseudocodul programului rucsac

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39 | \*) citește V,n, (p[i], v[i] cu i = 1,n)  \*) ordonează p[] și v[] în ordinea descrescătoare a raportului pi/vi  \*) x[i]=0, cu i = 1,n  i = 1  **vt** = pt = 0  **cât timp** vt < V **și** i ≤ n **execută**  **dacă** vt + v[i] ≤V **atunci**  x [i ] = 1  pt = pt+p[i]  vt = vt + v[k]  **altfel**  x[i]= (V-vt)/v[i]  vt=V  pt = pt + p[i] \* x[i]  @  i = i+1  @  **scrie** “Pretul total al obiectelor din rucsac=“, pt, “ Volumul ocupat=”, vt  **scrie** ”In rucsac s-au introdus:”  **pentru** i = 1,n **execută**  **dacă** x[i]>0 **atunci scrie ”-** obiectul **”,** i , “ (“, v [i]\* x[i], ”,” , p[i] „)”  @  **sfârșit** |

**Programul va afisa un meniu cu următoarele opțiuni:**

C - citire date de la tastatura

F - citire date dintr-un fisier al carui nume va fi preluat de la tastatura

A - afisare explicit date citite (Volum rucsac, nr. obiecte, ...)

R - rezolvare problema

I - info autor

T - termina programul

**Pentru primele 4 opțiuni se va implementa câte o funcție care să realizeze operația cerută.**

**TEMA DE REALIZAT ACASĂ:**

1. Se cere să se gaseasca o dispunere optimă a *n* fișiere cu lungimile L1, L2, .. ., Ln pe o bandă magnetică în ipoteza că timpul de citire al unui fișier este proporţional cu lungimea sa, iar pe bandă citirea fișierului k implică citirea celor k-1 fișiere precedente. Presupunem o distribuţie uniformă a frecvenţelor de citire a fișierelor.

Aranjarea optimă a fișierelor pe o bandă magnetică înseamnă obținerea unui timp mediu de citire minim. Structura fișierelor pe bandă



**Date de intrare**

L - lungimea benzii magnetice

n - numărul de fișiere ce trebuie aranjate

Nume\_Fișier\_1 - un șir de caractere ce conține numele primului fișier

Dimensiune\_Fișier\_1 - lungimea în octeți a informațiilor conținute în Fișier1

Nume\_Fișier\_2

Dimensiune\_Fișier\_2

...

...

Nume\_Fișier\_n

Dimensiune\_Fișier\_n

**Date de ieșire:**

Varianta 1: șirul (N1 L1) (N2 L2) ... (Nm Lm), unde cuplul (Ni Li) reprezintă numele și dimensiunea fișierului i.

Varianta 2: Permutarea p=(p1, p2, ... , pn), a lui (1,2,...,n) pentru care se obține minimul timpului mediu de citire, elementele pi reprezentând numerele de ordine ale fișierelor din introducerea datelor.

**Soluție propusă**

Se poate demonstra [1] că soluția optimă se obține atunci când fișierele sunt sortate în ordinea crescătoare a lungimii lor (lungime nume fișier + lungime informație).

Se va folosi funcția Greedy\_2 cu următoarele observații:

1. funcția PREL(A, n) va crea un vector V (cu Vi=Lungime\_nume\_fișier\_i + Lungime\_informație\_i) și va sorta crescător vectorul V

2. funcția POSIBIL(B,ai) va determina dacă lungimea bandei magnetice nu este depășită ()

**Bibliografie:**

1. S.-G. Pentiuc, Programarea Calculatoarelor și Limbaje de Programare II. Note de curs
2. Tutorial Python: http://www.tutorialspoint.com/python/