

# **Lucrare practică**

## **TEMA: Tehnica Greedy**

Efectua de : Grubleac  
Gabriela ,eleva clasei a 11-a

Verificat de : Guțu Maria ,  
profesor de informatică

## Cuprins

Descrierea aspectelor teoretice .....	2
Avantajele și dezavantajele metodei .....	2
Utilizarea metodei în rezolvarea problemelor .....	3
Concluzii .....	7
Bibliografie .....	7

### Descrierea aspectelor teoretice

Tehnica Greedy nu are o structura standard pentru toate tipurile de algoritm.

Mod de realizare:

La fiecare pas al unui algoritm greedy se adauga la solutie un element optim sau promitator (sau se “înghite”). Pentru implementare se considera o multime A, cu elemente de la 1 la n din care se aleg pe rand elemente ce indeplinesc criteriile de optimizare si se includ intr-o multime B (multimea solutiilor). Algoritmii greedy sunt performanti chiar daca problemele au dimensiuni mari. Tot timpul trebuie gasita multimea elementelor initiale ce se va optimiza in functie de criteriile.

În principiu, problemele de acest tip pot fi rezolvate prin metoda trierii, generînd consecutiv cele 2 la puterea n submulțimi  $A_i$  ale mulțimii A.

#### Avantajele și dezavantajele metodei

Dezavantajul metodei trierii constă în faptul că timpul cerut de algoritmii respectivi este foarte mare. Pentru a evita trierea tuturor submulțimilor  $A_i$ ,  $A_i \subseteq A$ , în metoda Greedy se utilizează un criteriu (o regulă) care asigură alegerea directă a elementelor necesare din mulțimea A. De obicei, criteriile sau regulile de selecție nu sînt indicate explicit în enunțul problemei și formularea lor cade în sarcina programatorului. În absența unor astfel de criterii metoda Greedy nu poate fi aplicată.

De asemenea Tehnica Greedy conduce la timp de calcul polinomial. Motivul care conduce la acest timp de calcul, tine de mecanismul tehnicii. Să presupunem că mulțimea din care se face alegerea are n elemente si că soluția are tot n elemente (caz maxim). Se fac n alegeri, la fiecare alegere se fac n teste, rezulta un algoritm cu timp  $O(n^2)$ .

Există câteva probleme celebre de algoritmică ce pot fi rezolvate cu metoda Greedy:

- Problema spectacolelor
- Problema continuă a rucsacului
- Algoritmul lui Dijkstra pentru determinarea drumurilor de cost minim într-un graf
- Algoritmul lui Prim și Algoritmul lui Kruskal pentru determinarea arborelui parțial de cost minim al unui graf

## Utilizarea metodei în rezolvarea problemelor

1. Scrieți un program care selectează elementele pozitive dintr-o mulțime A și le include în submulțimea B.

```
const nmax=1000;
var A : array [1..nmax] of real;
    n : 1..nmax;
    B : array [1..nmax] of real;
    m : 0..nmax;
    x : real;
    i : 1..nmax;
Function ExistaElemente : boolean;
var i : integer;{acesta functie accepta initial Solutia false; parcurgand
matricea A , dace sunt elemente pozitive Solutia devine true ;astfel se
poate incepe ciclul while din programul principal}
begin
    ExistaElemente:=false;
    for i:=1 to n do
        if A[i]>0 then ExistaElemente:=true;
    end;
procedure AlegeUnElement(var x : real);
var i : integer;{aceasta procedura extrage din matrice elementele pozitive
;pentru ca ciclul sa functioneze si ca sa nu sa se mai intoarca la aceleasi
valori se inlocuiesc elementele pozitive deja atasate cu 0}
begin
    i:=1;
    while A[i]<=0 do i:=i+1;
    x:=A[i];
    A[i]:=0;
end;
procedure IncludeElementul(x : real);
begin{aceasta procedura include elemental x in matricea B}
    m:=m+1;
    B[m]:=x;
end; begin
    write('Dați n='); readln(n);
    writeln('Dați elementele mulțimii A:');{n reprezinta numarul de element ce
se contin in multimea A}
    for i:=1 to n do read(A[i]);
    writeln;{se citesc elemntele multimii A}
```

```

m:=0;{myltimea Beste initial vida}
while ExistaElemente do
begin
  AlegeUnElement(x);
  IncludeElementul(x);
end;{aceasta este partea programului care reprezinta tehnica
Greedy;Aceasta functioneaza folosind un ciclu while ce se
opreste in momentul in care nu mai sunt numere positive ; ciclul
alege elementele pozitive din ciclu si le include in submultimea
B, conform subprogamelor anterioare }
writeln('Elementele multimii B:');
for i:=1 to m do writeln(B[i]);
readln;
end.

```

2. Scrieți un program, care afișează modalitatea de plată, folosind un număr minim de bancnote, a unei sume întregi S de lei ( $S < 20000$ ). Plata se efectuează folosind bancnote cu valoarea 1, 5, 10, 50, 100, 200 și 500 de lei. Numărul de bancnote de fiecare valoare se citește din fișierul text BANI.IN, care conține 7 rânduri, în fiecare din care sunt indicate numărul de bancnote respectiv de 1, 5, 10, 50, 100, 200 și 500 de lei.  
Intrare: Fișierul text BANI.IN și de la tastatură se citește suma S.  
Ieșire: Dacă e posibil de plătit această sumă S, atunci la ecran se va afișa valoarea bancnotei și numărul de bancnote respective utilizate la plată. Dacă bancnote de careva valoare nu se folosesc, atunci nu se afișează această valoare. Dacă nu este posibil de efectuat plata cu bancnotele indicate – afișați mesajul respectiv.

```

type tablou=array[1..3,1..7] of integer;
var s,ss,i : integer; a:tablou; f:text;
Procedure Afisare(sa:integer);var i:integer;
begin writeln('suma ',s);
if sa<>0
then writeln('nu poate fi transformata cu bancnotele date ')
else
begin writeln('se plateste cu urmatoarele bancnote');
for i:=1 to 7 do
if a[3,i]<>0
then writeln('bancnote de ',a[1,i]:6,' sau folosit ',a[3,i]);
end
end; { Afisare }
Procedure calcul(var sa:integer);
var nb:integer;
begin
i:=7;
while (i>=1) and (sa>0) do
begin nb:=sa div a[1,i];
if nb<>0 then if nb>= a[2,i]
then a[3,i]:=a[2,i]
else a[3,i]:=nb;

```

```

sa:=sa-a[3,i]*a[1,i];
i:=i-1;
end;
end; { calcul }
begin
a[1,1]:=1; a[1,2]:=5; a[1,3]:=10; a[1,4]:=50;
a[1,5]:=100; a[1,6]:=200; a[1,7]:=500;
assign (f,'bani.in'); reset(f);
for i:=1 to 7 do readln(f,a[2,i]);
write ('introduceti suma de lei S ');readln(s);
ss:=s; calcul(ss); Afisare(ss);

end.

```

3. O persoană are un rucsac cu care poate transporta o greutate maximă  $G$ . Persoana are la dispoziție  $n$  obiecte și cunoaște pentru fiecare obiect greutatea și câștigul care se obține în urma transportului său la destinație. Se cere să se precizeze ce obiecte trebuie să transporte persoana în așa fel încât câștigul să fie maxim

```

type tablou= array [1..100,1..100] of real;
matrice =array [1..100] of real ;
var a:matrice; c:tablou; f:text;
loc,n,g,i,j:integer; max,castig,dg:real;
begin
assign (f,'rucsac.txt'); reset (f);
readln(f,n,g);
for i:=1 to n do
begin readln(f,a[i,1],a[i,2]);
a[i,3]:=a[i,1]/a[i,2]; a[i,4]:=0;
end;
{sortam tabloul dupa eficienta}
for i:=1 to n-1 do
begin max:=a[i,3];loc:=i;
for j:=i+1 to n do
if a[j,3]>max then begin max:=a[j,3]; loc:=j; end;
c:=a[i]; a[i]:=a[loc]; a[loc]:=c;
end;
{Aflam cat din fiecare obiect se pune in rucsac si calculam castigul}
castig:=0;
i:=1; dg:=g;
writeln ('greutatea ','costul ','eficienta ','rucsac');
while (i<=n) and (dg>0) do
begin;
if dg>=a[i,2]
then begin castig:=castig+a[i,1];
dg:=dg-a[i,2]; a[i,4]:=1;
end
else begin castig:=castig+dg*a[i,3];
a[i,4]:=dg/a[i,2]; dg:=0;
end;
writeln (a[i,1]:6:2,a[i,2]:8:2,a[i,3]:12:2,a[i,4]:10:2);

```

```

i:=i+1;
end;
writeln ('greutatea rucsacului este ',g-dg:0:2);
writeln ('costul este ',castig:0:2);
end.

```

#### 4. Scrieti un program care calculeaza minimul din 3 numere prin tehnica Greedy

```

Program P1;
Var n, a1, a2, c:Integer;
Begin
read (n, a1,a2);
While n<>0 Do Begin
If (n>a1) Then a1:=n else {daca numarul n este mai mare decat primul cel
mai mare numar atunci maximul este n}
If (a2<a1) Then Begin
c:=a1;
a1:=a2;
a2:=c; end;
Readln (n); end;
Writeln (a1, ' ',a2);
end.

```

#### 5. Scrieti un program care calculeaza suma maxima o elementelor unei multimi .

```

const nmax=1000;
var A : array [1..nmax] of real;
n : 1..nmax;
B : array [1..nmax] of real;
m : 0..nmax;
x : real;
i : 1..nmax;
s: real;
Function ExistaElemente : boolean;
var i : integer;
begin
ExistaElemente:=false;
for i:=1 to n do
if A[i]>0 then ExistaElemente:=true;
end;
procedure AlegeUnElement(var x : real);
var i : integer;
begin
i:=1;
while A[i]<=0 do i:=i+1;
x:=A[i];
A[i]:=0;
end;
procedure IncludeElementul(x : real);
begin
m:=m+1;
B[m]:=x;
end;
begin
write('Dați n='); readln(n);

```

```

writeln('Dați elementele mulțimii A:');
for i:=1 to n do read(A[i]);
writeln;
m:=0;
while ExistaElemente do
begin
  AlegeUnElement(x);
  IncludeElementul(x);
end;

for i:=1 to m do s:=s+B[i];
writeln('suma maximala = ',s)
end.

```

## Concluzii

Tehnica Greedy este convenabil de a fi folosita doar in unele tipuri de probleme. In probleme ce necesita creerea unei submultimi, aplicarea acestei tehnici este mai convenabila decat aplicarea metodei trierii .In cazul in care nu exista un algoritm special de selectare a elementelor in submultime atunci acesta metoda nu poate fi folosita .

## Bibliografie

- <https://tpascalblog.wordpress.com>
- <http://dasinika.blogspot.com>
- <https://www.cyberforum.ru/pascalabc/thread2470480.html>
- Manualul de informatica clasa 11