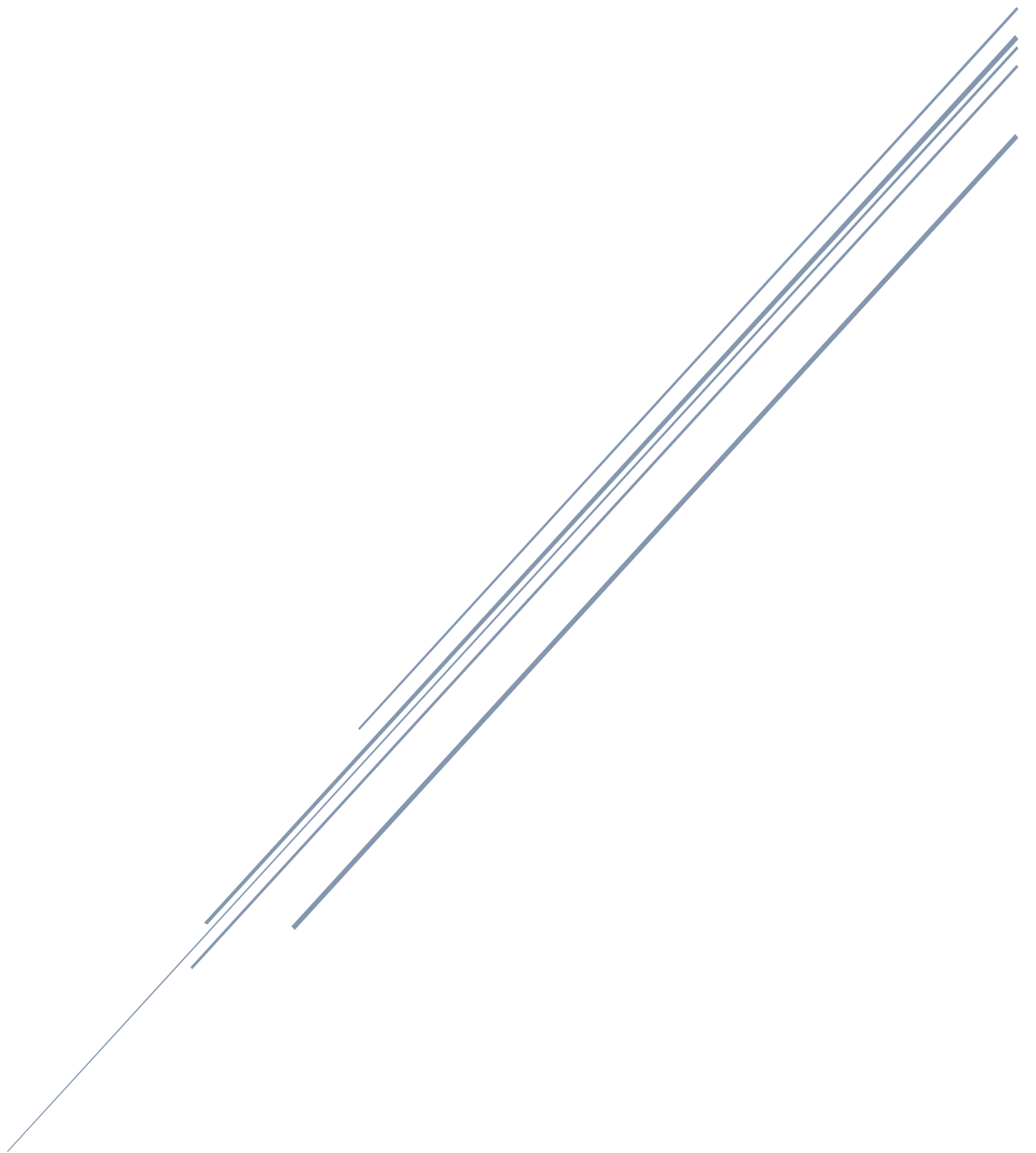


TEHNICA GREEDY



Tataru Vlada, clasa a XI-a "D"
12.05.2020

Cuprins

1. Tehnica Greedy. Noțiuni generale.	3
2. Structură. Schemă. Scop.....	4
3. Avantaje și dezavantaje.....	5
4. Pașii algoritmici metodei greedy.....	6
5. Probleme rezolvate.	7
6. Concluzie	12
7. Bibliografie.....	13

1. Tehnica Greedy. Noțiuni generale.

<https://www.slideshare.net/BalanVeronica/metoda-greedy1>

Metoda Greedy este una dintre cele mai directe tehnici de proiectare a algoritmilor care poate fi aplicată la o gamă largă de probleme. În general, această metodă se aplică problemelor de optimizare. Majoritatea acestor probleme constau în determinarea unei submulțimi B , a unei mulțimi A cu n elemente care să îndeplinească anumite condiții pentru a fi acceptată. Orice astfel de submulțime care respectă aceste restricții se numește soluție posibilă. Din mulțimea tuturor soluțiilor posibile se dorește determinarea unei soluții care maximizează sau minimizează o funcție de cost. O soluție posibilă care realizează acest lucru se numește soluție optimă. Considerăm că soluțiile posibile au următoarea proprietate: dacă B este o soluție posibilă, atunci orice submulțime a sa este soluție posibilă. Specificul acestei metode constă în faptul că se construiește soluția optimă pas cu pas, la fiecare pas fiind selectat (sau "înghițit") în soluție elementul care pare "cel mai bun" la momentul respectiv, în speranța că va duce la soluția optimă globală.

Algoritmii Greedy sunt caracterizați de metoda lor de funcționare: la fiecare pas se alege cel mai bun candidat posibil, după evaluarea tuturor acestora. Metoda determină întotdeauna o singură soluție, asigurând un optim local, dar nu întotdeauna și global. Tehnica Greedy este una de optimizare, rulând mai rapid decât un Backtracking, dar nefiind întotdeauna cea mai bună.

Când nu aveți o idee mai bună legată de o problemă, în timpul unui concurs, o implementare Greedy ar putea aduce în jur de 30% din punctaj.

În cele mai multe situații avem:

- O mulțime de candidați (valori de ales, noduri în graf etc.)
- O funcție care verifică dacă o mulțime de candidați constituie o soluție posibilă, nu neapărat optimă, a problemei
- O funcție care verifică dacă o mulțime de candidați este fezabilă, adică dacă este posibil să completăm această mulțime astfel încât să obținem o soluție posibilă (nu neapărat optimă).
- O funcție de selecție ce indică într-un anumit moment care este cel mai potrivit dintre candidații nefolosiți.
- O funcție obiectiv care dă valoarea unei soluții (timpul necesar executării tuturor lucrărilor într-o anumită ordine), lungimea drumului pe care l-am găsit etc). [1]

2. Structură. Schemă. Scop.

<https://www.slideshare.net/BalanVeronica/tehnica-greedy>

Această metodă presupune că problemele pe care trebuie să le rezolvăm au următoarea structură:

- se dă o mulțime $A = \{a_1, a_2, \dots, a_n\}$ formată din n elemente;
- se cere să determinăm o submulțime B , $B \subset A$, care îndeplinește anumite condiții pentru a fi acceptată ca soluție.

În principiu, problemele de acest tip pot fi rezolvate prin metoda trierii, generând consecutiv cele 2^n submulțimi A_i ale mulțimii A . Pentru a evita trierea tuturor submulțimilor A_i , $A_i \subset A$, 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. Evident, în absența unor astfel de criterii metoda *Greedy* nu poate fi aplicată. Schema generală a unui algoritm bazat pe metoda *Greedy* poate fi redată cu ajutorul unui ciclu: [2]

```
While  
ExistaElemente do  
    Begin  
        AlegeUnElement(x);
```

Functia **AlegeElementul**, extrage din, multimea A un alt X .

Functia **IncludeElementul**, inscrie elementul selectat in submultimea B .

Scop- indentificarea problemelor în care soluțiile optimă este o submulțime inclusă într-o submulțime de dată, care trebuie să îndeplinească anumite condiții.

3. Avantaje și dezavantaje

https://prezi.com/ys_iuqvvoxsi/tehnica-greedy/

Tehnica Greedy este una de optimizare, rulând mai rapid, dar nefiind întotdeauna cea mai bună. Algoritmii Greedy nu conduc în mod necesar la o soluție optimă. Și nici nu este posibilă formularea unui criteriu general conform căruia să putem stabili exact dacă metoda Greedy rezolvă sau nu o anumită problemă de optimizare. Din acest motiv, orice algoritm Greedy trebuie însoțit de o demonstrație a corectitudinii sale. Demonstrația faptului că o anumită problemă are proprietatea alegerii Greedy se face de obicei prin inducție matematică. Avantajul timpului polinomial, conduce la necesitatea utilizării tehnicii Greedy. Pentru problemele pentru care nu se cunosc algoritmi care necesită timp polinomial, se caută soluții, chiar dacă nu optime, dar apropiate de acestea și care au fost obținute în timp util. Multe din aceste soluții sunt obținute cu Greedy. De asemenea, Tehnica Greedy poate fi aplicată multor probleme, iar soluția se construiește progresiv, pas cu pas. [3]

4. Pașii algoritmici metodei greedy

<https://www.slideshare.net/LuminiaMihailov/metoda-greedy-47870787>

Pasul 1- se inițializează mulțimea S cu mulțimea vidă;

Pasul 2 –se alege din mulțimea A elementului a care este candidatul optim al soluției;

Pasul 3- se elimină elementul a din mulțimea A

Pasul 4 – dacă el poate fii elemnent al soluției,atunci elementul a se adaugă la mulțimea S;

Pasul 5- dacă mulțimea S este soluția problemei,atuni se afisează soluția.astfel se afisează mesajul “Nu s-a găsit soluția”. [4]

5.Probleme rezolvate.

<http://caterinamacovenco.blogspot.com/p/metoda-greedy.html>

<https://www.slideshare.net/BalanVeronica/metoda-greedy1>

<http://ctice.md/public/download.php?id=75>

https://www.slideshare.net/yoanna_ioana/problema-rucsacului-presentation-948687

<https://tpascalblog.wordpress.com/>

1. Se dă un rucsac de o anumită capacitate, greutate și un număr de n obiecte specificându-se masa obiectelor. Se cere un program care să determine variantă de introducere a obiectelor în rucsac astfel încât să încapă cât mai multe obiecte. [5]

```
Program Greedy;
Var g:array [1..10] of integer;
i,n,Gm,R, aux : integer;
ok:boolean;
begin
  writeln('nr obiecte'); readln(n);
  writeln('capacitate rucsac'); readln(R);
  writeln(' Obiectele de luat în rucsac:' );
  for i:=1 to n do
    read (g[i]);
  ok:=false;
  while(ok=false) do
    begin
      ok:=true;
      for i:=1 to n-1 do
        if g[i]>g[i+1] then
          begin
            aux:=g[i];
            g[i]:=g[i+1];
            g[i+1]:=aux;
            ok:=false;
          end;
      end;
      writeln;
      for i:=1 to n do write( g[i], '*');
      Gm:=0 ;
      i:=1;
      while ( Gm +g[i]<=R ) do
        begin
          Gm:=Gm+g[i];
          i:=i+1;
        end;
      writeln('sunt' ,i-1,' obiecte cu greutate' , Gm,') ;
      writeln ( ' a ramas' , R-Gm,' loc liber' ) ;
```

```
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. [1]

```
Program bani;  
type tablou=array[1..3,1..7] of integer;  
var s,ss,i : integer;  
a:tablou;  
f:text;  
Procedure Afisare(sa: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;  
  
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;  
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. Se consideră mulțimea $A=\{a_1, a_2, \dots, a_i, \dots, a_n\}$ elementele căreia sînt numere reale, iar cel puțin unul din ele satisface condiția $a_i > 0$. Elaborați un program care determină o submulțime $B, B \subseteq A$, astfel încît suma elementelor din B să fi e maximă. [6]

```

Program P153;
{ Tehnica Greedy }
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;
i:=1;
while A[i]<=0 do i:=i+1;
x:=A[i];
A[i]:=0;
end; { AlegeUnElement }
procedure IncludeElementul(x : real);
begin
m:=m+1;
B[m]:=x;
end; { IncludeElementul }
begin
write('Dați n='); readln(n);
writeln('Dați elementele mulțimii A:');
for i:=1 to n do read(A[i]);
writeln;
m:=0;
begin
ExistaElemente:=false;
for i:=1 to n do
if A[i]>0 then ExistaElemente:=true;
end; { ExistaElemente }
procedure AlegeUnElement(var x : real);
var i : integer;
begin
while ExistaElemente do
begin
AlegeUnElement(x);
IncludeElementul(x);
end;
writeln('Elementele mulțimii B:');
for i:=1 to m do writeln(B[i]);
readln;
end.

```

4. O persoană are un rucsac cu ajutorul căruia 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. [7]

```
Program rucsac;
Type obiect=record
C,Go,E:real;
end ;
var ok:boolean;
v:array[1..100] of obiect;
aux:obiect
i,n,t:integer;
G, ct:real;
Begin
write('n='); readln(n);
write('G='); readln(G);
For i:=1 to n do Begin
write('castigul pentru obiectul', i);
readln(v[i].c);
write('greutatea obiectului ',i);
readln(v[i].Go);
v[i].e:=v[i].c/v[i].Go;
End;
t:=1;
Repeat
ok:=true;
for i:=1 to n-t do
if v[i].e<v[i+1].e then Begin
ok:=false;
aux:=v[i];
v[i]:=v[i+1];
v[i+1]:=aux;
end;
t:=t+1;
until ok;
i:=1; ct:=0;
while (G<>0) and (i<=n) do Begin
if v[i].Go<G then Begin
ct:=ct+v[i].c;
G:=G-v[i].Go;
writeln('obiectul',i,'1');
end
else begin
P:=G*100/v[i].c*p)/100;
writeln('obiectul,i,se introduce in procent de ',P);
end;
i:=i+1;
end;
writeln('castigul total este ',ct);
end;
```

```
readln;  
End.
```

```
5. Program Maxim;  
   Var n, a1, a2, c:Integer;  
Begin  
  a1:=-MAXINT; (initializam primele 2 numere si n cu o constanta predefinita)  
  a2:=-MAXINT;  
  n:=-MAXINT;  
  While n<>0 Do Begin  
    If (n>a1) Then a1:=n; (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; (interschimbare)  
    Readln (n); end;  
    Writeln ('a1, ',a2');  
  End. [8]
```

6. Concluzie

Metoda Greedy este o metodă foarte eficientă în cazul când dorim să aflăm optim în cât mai scurt timp posibil, deoarece algoritmi sunt polinomiali. Această metodă nu poate fi aplicată la toate problemele. Ea poate fi aplicată doar atunci când din enunțul problemei poate fi dedusă regula care asigură selecția direct a elementelor necesare din mulțimea dată. Rezultatul unui algoritm greedy pentru o problemă dată depinde și de datele concrete ale problemei, sau chiar de ordinea introducerii lor. Însă folosirea acestei metode neștiind alta, este o cale bună de a rezolva un program.

7. Bibliografie

<https://www.slideshare.net/BalanVeronica/metoda-greedy1> [1]

<https://www.slideshare.net/BalanVeronica/tehnica-greedy> [2]

https://prezi.com/ys_iuqvyoysi/tehnica-greedy/ [3]

<https://www.slideshare.net/LuminiaMihailov/metoda-greedy-47870787> [4]

<http://caterinamacovenco.blogspot.com/p/metoda-greedy.html> [5]

<http://ctice.md/public/download.php?id=75> [6]

https://www.slideshare.net/yoanna_ioana/problema-rucsacului-presentation-948687 [7]

<https://tpascalblog.wordpress.com/> [8]