I.P.L.T Spiru Haret

Referat

Tema: “Metoda trierii”

Elev cl a XI-a “D”: Bondari Sofia

Profesor: Guțu Maria

Chișinău 2020

Cuprins

Cuprins

1.Aspecte teoretice 3

1.1 Ce este metoda trierii 3

1.2 Necesitatea studierii metodei 3

1.4 Avantaje și dezavantaje 4

2. Exemple de probleme 5

3.Concluzie 9

1.Aspecte teoretice

1.1 Ce este metoda trierii

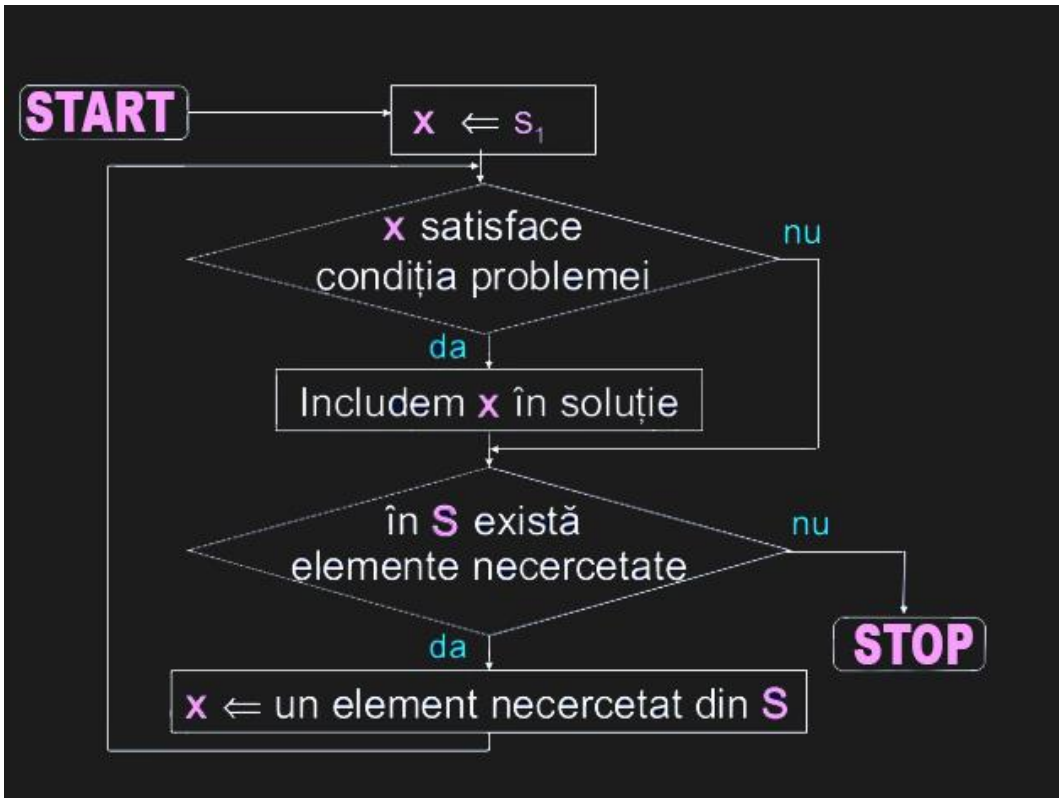
Se numeşte **metoda trierii** o metodă ce indentifică toate soluţiile unei probleme în dependenţă de mulţimea soluţiilor posibile. Toate soluţiile se identifică prin valori, ce aparţin tipurilor de date studiate: integer, boolean, enumerare, char, subdomeniu, tablouri unidimensionale.

Fie P o problemă, soluţia căreia se află printre elementele mulţimii S cu un număr finit de elemente. S={s1, s2 , s3 , ... , sn} . Soluţia se determină prin analiza fiecărui element si din mulţimea S.

1.2 Necesitatea studierii metodei

Metoda trierii trebuie să fie studiată la informatică în contextul studierii algoritmilor, deoarece programele respective sunt relativ simple, iar depanarea lor nu necesită teste sofisticate. Complexitatea temporală a acestor algorimi este determinată de numărul de elemente k din mulţimea soluţiilor posibile S. În majoritatea problemelor de o reală importanţă practică metoda trierii conduce la algoritmiii exponenţiali. Întrucît algoritmii exponenţiali sunt inacceptabili în cazul datelor de intrare foarte mari,metoda trierii este aplicată numai în scopuri didactice sau pentur elaborarea unor programe al căror timp de execuţie nu este critic. [1]

1.3 Schema de aplicare



[2]

1.4 Avantaje și dezavantaje

Avantaje	Dezavantaje
• *Programele respective sunt relative simple, iar depanarea lor nu necesită teste sophisticate.* • *Problemele relative simple sunt effectuate rapid, încadrânu-se în timpul minim de execuție.*	• *Metoda trierii nu poate fi aplicată în problemele complexe ce necesită date de intrare ale căror valori sunt foarte mari.* • *Pentru verificarea tuturor cazurilor va fi necesar un timp mare de execuție care depinde de nr k de elemente ce trebuie găsite în mulțimea soluțiilor posibile S.* • *Întrucât algoritmii exponențiali sunt inacceptabili în cazul datelor de intrare foarte mari ,metoda trierii este aplicată numai în scopuri didactice sau pentru elaborarea unor programe ale căror timp de execuție nu este critic.*

[2],[3]

2. Exemple de probleme

2.1 Se consideră numerele naturale din mulţimea {0, 1, 2, …, n}. Elaboraţi un program care determină pentru cîte numere K din această mulţime suma cifrelor fiecărui număr este egală cu m. În particular, pentru n=100 și m=2, în mulţimea{0, 1, 2, …, 100} există 3 numere care satisfac condiţiile problemei: 2, 11 și 20. Prin urmare, K=3.

Rezolvare:

```
Type Natural=0..MaxInt;
Var I, k, m, n : Natural;
Function SumaCifrelor(i:Natural): Natural;
Var suma: Natural;
Begin
Suma:=0;
Repeat
Suma:=suma+(I mod 10);
i:=i div 10;
until i=0;
SumaCifrelor:=suma;
End;
Function SolutiePosibila(i:Natural):Boolean;
Begin
If SumaCifrelor(i)=m then SolutiaPosibila:=true
Else SolutiePosibila:=false;
End;
Procedure PrelucrareaSolutiei(i:Natural);
Begin
Writeln('i=', i);
K:=k+1;
End;
Begin
Write('Dati n=');
readln(n);
Write('Dati m=');
readln(m);
K:=0;
For i:=0 to n do
If SolutiePosibila(i) then PrelucrareaSolutiei(i);
Writeln('K=', K);
Readln;
End.
```

2.2 Prooblema Rucsac

```
Program Rucsac;
type tablou=array [1..4] of real;
matrice=array [1..10] of tablou;
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;
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;
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.
```

2.3. Problema Spectacolelor.

```
Program Spectacole;
ora_inc, ora_sf:integer;
ord:integer;
end;
Var v:array[1..30] of spectacol;
n, ultim, nr:integer;
procedure sortare;
```

```
var i,j :integer; aux:spectacol;
begin
for i:=1 to n-1 do
for j:=i+1 to n do
if v[j].ora_sf < v[i].ora_sf then
begin aux:=v[j];v[j]:=v[i];v[i]:=aux;end;
end;
procedure citire;
var hh, mm, i:integer;
begin
write(‘Numarul de spectacole:’); readln(n);
for i:=1 to n do
begin
write(‘Spectacolul, i, incepe la:’); readln(hh,mm);
v[i]. ora_inc:=hh*60+mm;
write (‘Spectacolul, i, se termina la:’);
readln(hh,mm); v[i].ora_sf:=hh*60+mm;v[i].ord:=i;
end;
end;
Type spectacol=record
var ;integer;
begin
writeln(’Ordinea spectacolelor este:’);
ultim:=1;nr:=1; write(v[1].ord,’ ’);
for i:=2 to n do
if v[i].ora_inc>v[ultim].ora_sf then begin
write(v[i].ord,’ ’);ultim:=i;Inc(nr);
end;
writeln(‘Se pot juca ’, nr, ‘ spectacole’);
end;
begin
citire; sortare; greedy;
end.
```

2.4. Problema Tablou

```
Program Tablou;
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.
```

2.5. Problemă suma maximă

```
Program suma_maxima;
var s,x:array[1..20] of real;
i,k,n:integer;
begin
write('Numarul de elemente n = ');
readln(n);
for i:=1 to n do begin
write('x[',i,']= ');
readln(x[i])
end; k:=0;
for i:=1 to n do
if x[i]>0 then
begin
k:=k+1; s[k]:=x[i]
end;
for i:=1 to k do write(s[i]:5:2,' ');
readln;
end.
```

3.Concluzie

Avantajul principal al algoritmilor bazați pe metoda trierii constă în faptul că programele respective sunt relativ simple, iar depanarea lor nu necesită teste sofisticate. În majoritatea problemelor de o reală importanță practică metoda trierii conduce la algoritmii exponențiali. Întrucît algoritmii exponențiali sunt inacceptabili în cazul datelor de intrare foarte mari, metoda trierii este aplicată numai în scopuri didactice sau pentru elaborarea unor programe al căror timp de execuție este critic. De obicei, algoritmii bazați pe metoda Greedy sunt algoritmi polinomiali.

Bibliografia

1. http://timofti7.simplesite.com/435052344
2. https://www.mindmeister.com/689967199/metoda-trierii
3. http://blogoinform.blogspot.com/p/metoda-trierii.html