

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

4/21/2020

METODA TRIERII

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and curve upwards and to the right.

Bîrsa Gheorghe

Cuprins

1. Aspecte teoretice.....	2
1.1. Definitie	2
1.2. Schema generală	2
1.3. Operatii realizate.....	3
2.Probleme rezolvate	3
3.Probleme din cotidian	8
Probleme din cotidian care pot fi rezolvate utilizând această metodă:.....	8
4.Conluzie(Avantaje/Dezavantaje)	8
5. BIBLIOGRAFIE	9

1. Aspecte teoretice

1.1. Definitie

Se numește metoda trierii o metodă ce indentifică toate soluțiile unei probleme în dependență de mulțimea soluțiilor posibile. Toate soluțiile se identifică prin valori, ce aparțin tipurilor de date studiate: integer, boolean, enumerare, char, subdomeniu, tablouri unidimensionale.

Fie P o problemă, soluția căreia se află printre elementele mulțimii S cu un număr finit de elemente. $S = \{s_1, s_2, s_3, \dots, s_n\}$. Soluția se determină prin analiza fiecărui element si din mulțimea S . [1]

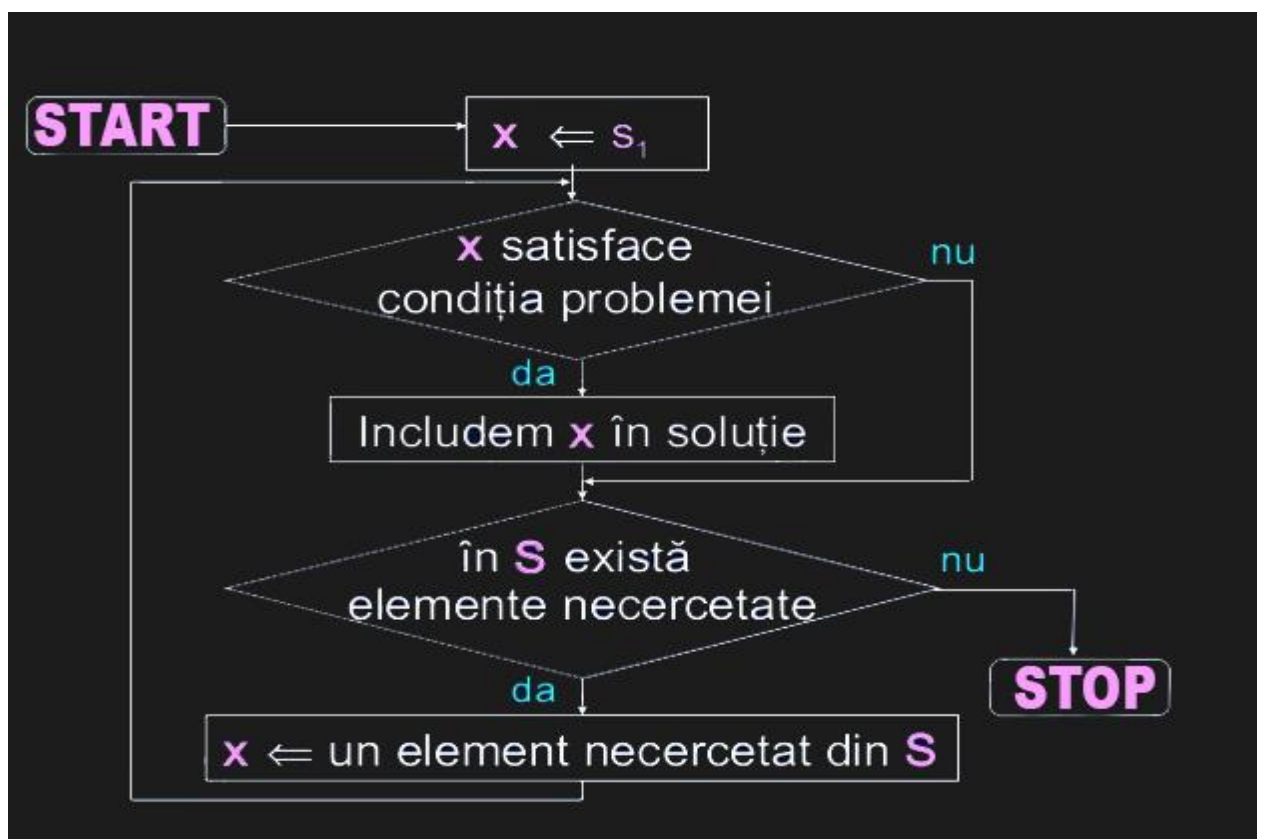
1.2. Schema generală

```
for i:=1 to k do
```

```
if SolutiePosibila ( $S_i$ ) then PrelucareaSolutiei ( $S_i$ )
```

Unde $SolutiePosibila$ este o functie booleana care returneaza valoarea true daca elemental S_i satisface conditiile problemei si false in caz contrarm iar $PrelucrareaSolutiei$ este o procedura care efectueaza prelucraea elementului selectat. [5]

Schema de aplicare a metodei trierii:



[2]

1.3. Operatii realizate

Generarea soluțiilor posibile necesită elaborarea unor algoritmi speciali. În general, acești algoritmi realizează operațiile legate de prelucrarea unor mulțimi:

- reuniunea;
- intersecția;
- diferența;
- generarea tuturor submulțimilor;
- generarea elementelor unui produs cartezian;
- generarea permutărilor, aranjamentelor sau combinărilor de obiecte etc.[4]

2.Probleme rezolvate

```
✓ Program P1;  
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;
```

Se consideră numerele naturale din mulțimea $\{0, 1, 2, \dots, n\}$. Elaboram un program care determină pentru câte numere K din această mulțime suma cifrelor fiecărui număr este egală cu m . În particular, pentru $n=100$ și $m=2$, în mulțimea $\{0, 1, 2, \dots, 100\}$ există 3 numere care satisfac condițiile problemei: 2, 11 și 20. Prin urmare, $K=3$.

```

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.

```

```

    ✓ Program P2; { Suma cifrelor unui număr natural }
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; { SumaCifrelor }
function SolutiePosibila(i:Natural):boolean;
begin
    if SumaCifrelor(i)=m then SolutiePosibila:=true else SolutiePosibila:=false;
end; { SumaCifrelor }
procedure PrelucrareaSolutiei(i:Natural);
begin
    writeln('i=', i);
    K:=K+1;
end; { PrelucrareaSolutiei }
begin

```

```

write('Dați n='); readln(n); write('Dați m='); readln(m);
K:=0;
for i:=0 to n do
if SolutiePosibila(i) then PrelucrareaSolutiei(i);
writeln('K=', K);
readln;
end.

```

```

✓ Program P3; { Puncte pe un plan euclidian }
const nmax=30;
type Punct = record
x, y : real;
end;
Indice = 1..nmax;
var P : array[Indice] of Punct;
j, m, n : Indice; dmax : real; { distanța maxima } PA, PB : Punct;
function Distanta(A, B : Punct) : real;
begin
Distanta:=sqrt(sqr(A.x-B.x)+sqr(A.y-B.y));
end; { Distanta }
function SolutiePosibila(j,m:Indice):boolean;
begin
if j<>m then SolutiePosibila:=true else
SolutiePosibila:=false;
end; { SolutiePosibila }
procedure PrelucrareaSolutiei(A, B : Punct);
Begin
if Distanta(A, B)>dmax then
begin PA:=A; PB:=B; dmax:=Distanta(A, B);
end;
end; { PrelucrareaSolutiei }
begin

```

```

write('Dati n='); readln(n); writeln('Dați coordonatele x, y ale punctelor');
for j:=1 to n do begin
write('P[', j, ']: '); readln(P[j].x, P[j].y);
end;
dmax:=0;
for j:=1 to n do for m:=1 to n do
if SolutiePosibila(j, m) then
PrelucrareaSolutiei(P[j], P[m]);
writeln('Soluția: PA=(', PA.x:5:2, ', ', PA.y:5:2, ')');
writeln(' PB=(', PB.x:5:2, ', ', PB.y:5:2, ')');
readln;
end.

```

✓ Program P4;

```

var i,k,n,m:integer;
function NrPrim(a:integer):boolean;
Var i:integer;
t:boolean;r:real;
begin
t:=true;
r:=sqrt(a);
i:=2;
while (i<=r) and t do begin if (a mod i)=0 then t:=false;
i:=i+1;
end;
NrPrim:=t;
end;
function SolutiePosibila (i:integer):boolean;
begin
SolutiePosibila:=NrPrim(n);
end ;
procedure PrelucrareaSolutiei(i:integer);
begin
writeln('i=',i);
k:=k+1;
end;
begin
writeln('Dati n=');readln(n);
writeln('Dati m=');readln(m);

```

Se considera numerele naturale din multimea $\{0,1,2,\dots,n\}$.Elaboram un program care determina cate numere prime sunt mai mari decat numarul natural dat m.

```

k:=0;
writeln('raspuns');
if NrPrim(n) then writeln('numarul ',n,' este prim') else writeln('numarul',n,' nu
este prim');
for
i:=n+1 to m do if
NrPrim(i)=true and (i<m) then
PrelucrareaSolutiei(i);
writeln('k=',k);
end.

```

```

✓ Program P5;
var n,i,m,k,x,sum:integer;
function nrperfect(x:integer):boolean;
begin
i:=1; sum:=0;
repeat
if (x mod i)=0 then sum:=sum+i;
i:=i+1
until i>(x div 2);
if sum=x then nrperfect:=true else nrperfect:=false;
end;
procedure prelucrareasolutiei(x:integer);
begin
writeln(x);
k:=k+1; end;
begin
k:=0;
readln(m);
for x:=1 to m-1 do
if nrperfect(x) then prelucrareasolutiei(x);
writeln('k=',k);
end.

```

Un numar se numeste perfect
daca este egal cu suma
divizorilor lui, in afara de el
insusi. Se afla numerele
perfecte mai mici decat
numarul natural dat.

3.Probleme din cotidian

Probleme din cotidian care pot fi rezolvate utilizând această metodă:

- aflarea numărului minim de monede care pot fi date drept plată sau rest;
- medicii deseori se confruntă cu necesitatea aplicării metodei trierii cazurilor, când numărul răniților sau bolnavilor este foarte mare, medicul fiind suprasolicitat, în cazul unui război, sau când își periclitizează propria viață în cazul unei epidemii periculoase;
- aflarea ariei maxime a unui lot de teren, avînd la dispoziție o anumită lungime de sîrmă ghimpată, spre exemplu (ca perimetru dat);
- generarea submulțimilor unei mulțimi (aflarea tuturor combinațiilor posibile), ceea ce ne poate fi foarte util în viața de zi cu zi;
- afișarea coordonatelor a două puncte date ce au distanță minimă sau maximă, ceea ce va fi foarte folositor dacă plănuim o călătorie;
- calcularea șanselor de a lua premiul mare la loterie etc.[3]

4.Concluzie(Avantaje/Dezavantaje)

Avantajul principal al algoritmilor bazați pe metoda trierii constă în faptul că programele respective sînt relativ simple, iar depanarea lor nu necesită teste sofisticate. Complexitatea temporală a acestor algoritmi este determinată de numărul de elemente k din mulțimea soluțiilor posibile S . În majoritatea problemelor de o reală importanță practică metoda trierii conduce la algoritmi exponențiali.

Dezavantajul ar fi ca algoritmi exponențiali sînt 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 nu este critic.

5. BIBLIOGRAFIE

<https://www.slideshare.net/foegirl/metoda-trierii-33371122> [1]

<http://caterinamacovenco.blogspot.com/p/tehnici-de-programare.html> [2]

<http://timofti7.simplesite.com/435052344> [3]

<http://blogoinform.blogspot.com/p/metoda-trierii.html> [4]

Manual de informatica clasa a 11-a Autor Anatol Gremalschi[5]