# Lucrare Științifică

# Tema: *Metoda trierii*

Data: 21.04.2020

Autor: Stegărescu Olivia cl.11C

Profesor: Guțu Maria

# Definiție

Se numește metoda trierii metoda ce identifică toate soluțiile unei probleme in dependență de mulțimea soluțiilor posibile. Toate soluțiile se indentifică prin valori, ce aparțin tipurilor de date studiate: integer, boolean, enumerare sau subdomeniu.

Metoda trierii este 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.

SCHEMA GENERALĂ

for i:=1 to k do

if SolutiePosibila (si) then PrelucrareaSolutiei (si)

(SolutiePosibila este o funcţie booleana care returneaza valoarea true dacă elementul si satisface condiţiile problemei şi false în caz contrar, iar PrelucrareaSolutiei este o procedură care efectuează prelucrarea elementului selectat. De obicei, în această procedură soluţia si este afişată la ecran.)

Generarea soluţiilor posibile necesită elaborarea unor algoritmi speciali. În general,aceşti algoritmi realizează operaţiile legate de prelucrarea unor mulţimi:

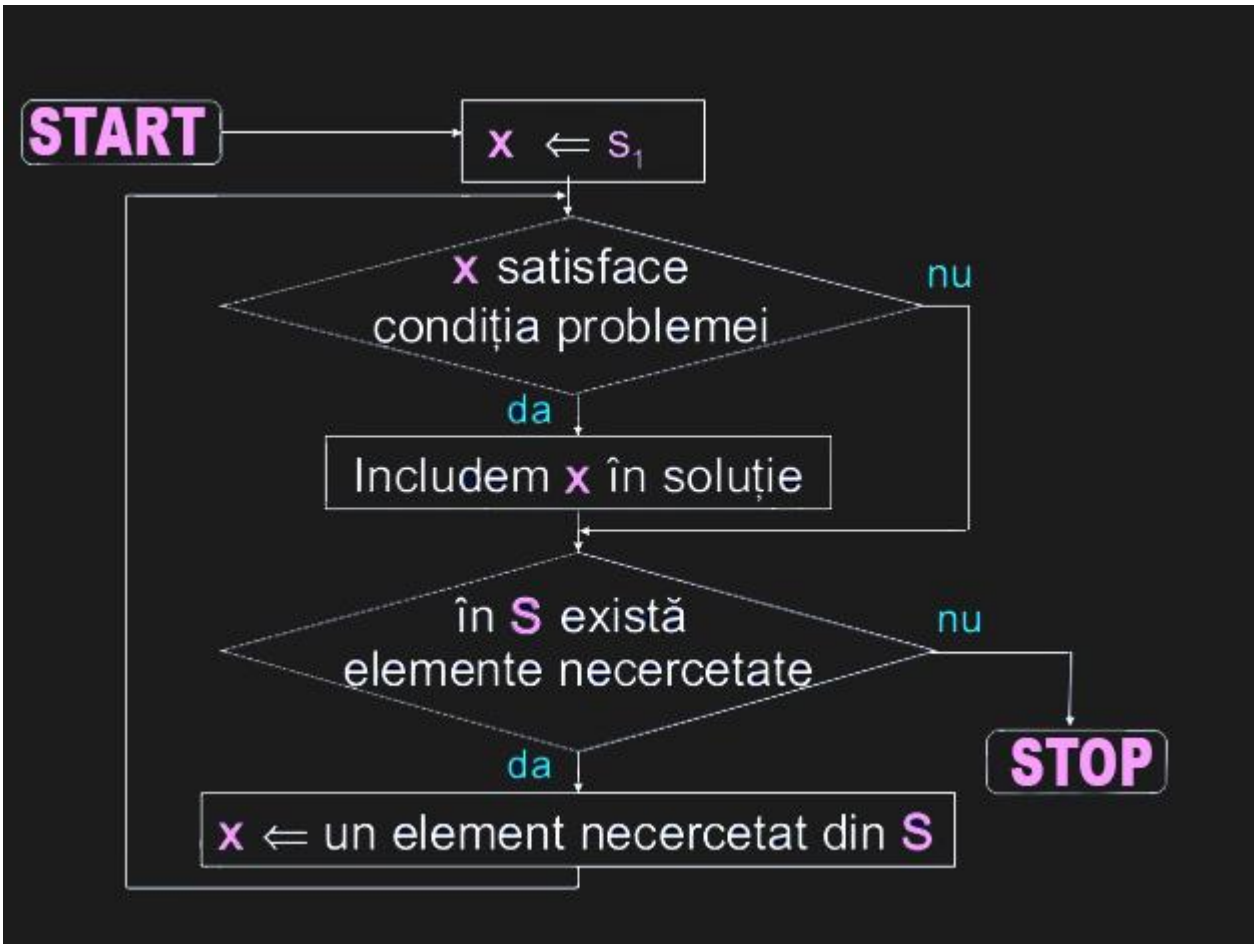
- - reuniunea;
- - intersecţia;
- - diferenţa;
- - generarea tuturor submulţimilor;
- - generarea elementelor unui produs cartezian;
- - generarea permutărilor, aranjamentelor sau combinărilor de obiecte etc.

Avantajul principal al algoritmilor bazaţi pe metoda trierii constă în faptul că 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.

În probleme mai complicate este nevoie de a reprezenta aceste elemente prin tablouri, articole sau mulțimi.

Schema de aplicare a metodei trierii este reprezentată mai jos:



## Probleme din viață rezolvabile prin metoda trierii

Metoda trierii poate fi folosită pentru rezolvarea următoarelor probleme din viață:

• 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 periclitează 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.

**Exemplu de problemă 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 si m=2, în mulţimea{0, 1, 2, …, 100} există 3 numere care satisfac condiţiile problemei: 2, 11 si 20.Prin urmare, K=3.

**Rezolvare:**

```
Program Pex;
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.
```

Exemplul 2.

Se consideră numerele naturale din mulţimea {0, 1, 2, ..., n}. Elaboraţi un program care determină pentru cîte numere K din această mulţime suma cifrelor fi ecărui număr este egală cu m. În particular, pentru n = 100 şi m = 2, în mulţimea {0, 1, 2, …, 100} există 3 numere care satisfac condiţiile problemei: 2, 11 şi 20. Prin urmare, K = 3. Rezolvare. Evident, mulţimea soluţiilor posibile S = {0, 1, 2, …, n}. În programul ce urmează suma cifrelor oricărui număr natural i, i∈S, se calculează cu ajutorul funcţiei SumaCifrelor. Separarea cifrelor zecimale din scrierea numărului natural i se efectuează de la dreapta la stînga prin împărţirea numărului i şi a cîturilor respective la baza 10.

```
Program P151;

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

Exemplul 3 .

Se consideră mulţimea P = {P1, P2, …, Pn} formată din n puncte ($2 \le n \le 30$) pe un plan euclidian. Fiecare punct Pj este defi nit prin coordonatele sale xj , yj . Elaboraţi un program care afişează la ecran coordonatele punctelor Pa, Pb distanţa dintre care este maximă. Rezolvare. Mulţimea soluţiilor posibile S = P×P. Elementele (Pj , Pm) ale produsului cartezian P×P pot fi generate cu ajutorul a două cicluri imbricate:

```
Program P152;
{ 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;
{ distanţa 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('Daţi 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('Soluţia: PA=(', PA.x:5:2, ',', PA.y:5:2, ')');
writeln(' PB=(', PB.x:5:2, ',', PB.y:5:2, ')');
readln;
end.
```

## 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.

# Bibliografie:


- http://caterinamacovenco.blogspot.com/p/tehnici-de-programare.html
- file:///D:/Downloads/XI_Informatica%20(in%20limba%20romana).pdf
- http://blogoinform.blogspot.com/p/metoda-trierii.html