Metoda trierii

Ciumașu Galina
cl. a XI-a ”D”
IPLT ”Spiru Haret”

Cuprins

Definiție [1]

Metoda trierii este o metodă ce indentifică toate soluțiile unei probleme în dependență de mulțimea soluțiilor posibile. Toate soluțiile se indentifică prin valori, ce aparțin tipurilor de date studiate: integer, boolean, enumerare, char, subdomeniu, tablouri unidimensionale. În problemele mai complicate este nevoie de a reprezenta aceste elemente prin tablouri, articole, mulțimi.

Necesitate [1]

Metoda Trierii este aplicată pe larg în soluţionarea problemelor avînd scopuri didactice. Aceasta însa nu poate fi aplicată problemelor complexe ce necesită date de intrare a căror valori sunt foarte mari. Astfel de date ce sînt supuse prelucrării conduc spre algoritmi exponenţiali. Pentru verificarea tuturor cazurilor va fi necesar de un timp mare de execuţie care depinde de nr. de k elemente ce trebuie găsite in mulţimea soluţiilor posibile S. Însa timpul pentru majoritatea execuţiilor în ziua de azi este limitat de diferite circumstanţe de aceea metoda trierii se foloseste numai în cazul cînd timpul de execuţie nu este critic.

De ce avem nevoie de această metodă?: [1]

1. Metoda trierii cercetează toate cazurile posibile introduse asftel selectând soluțiile care ar îndeplini condiția problemei.
2. Datorită acestei structuri de soluționare, probleme relativ simple sunt efectuate rapid, încadrându-se în timpul minim de execuție.

Avantaje [1]

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. 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 al căror timp de execuție este critic.

Metode de studiere [1]

Toate soluțiile se indentifică prin valori, ce aparțin 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.

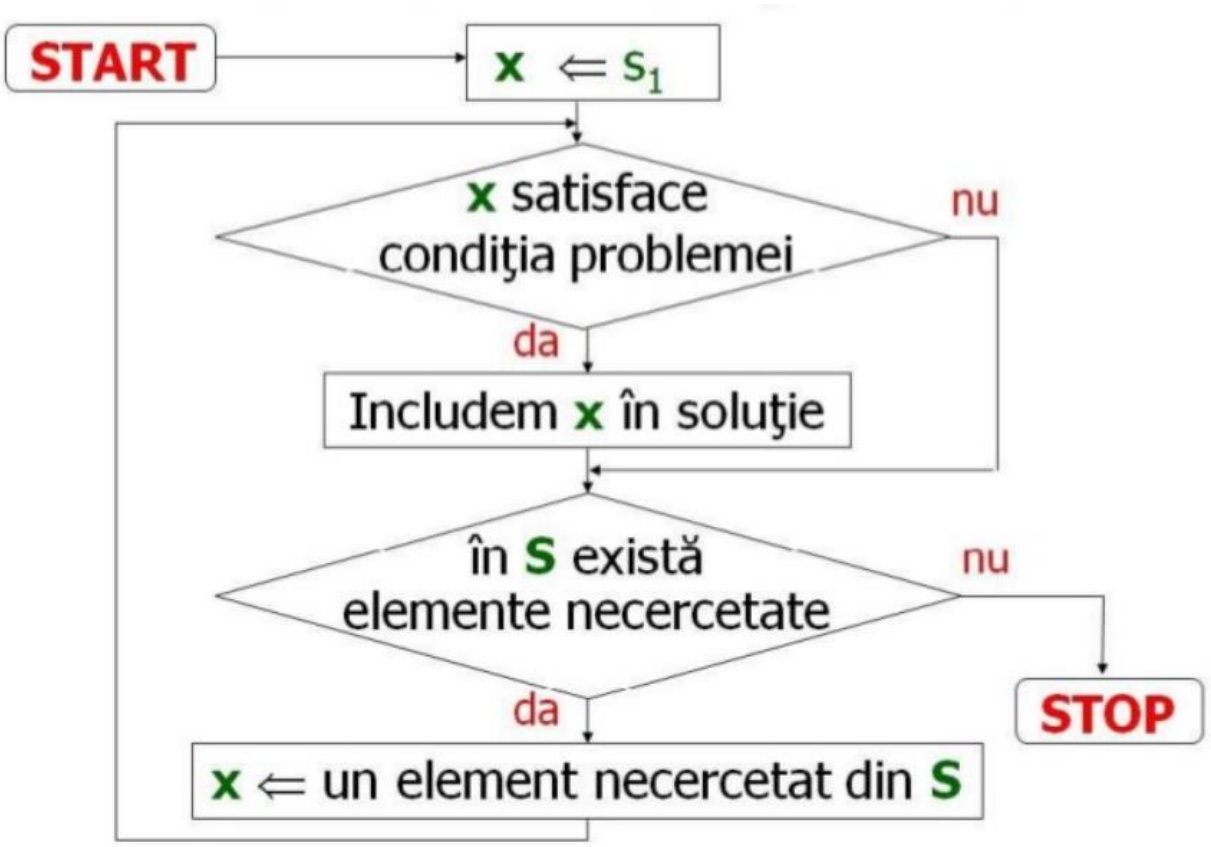
Schema generală:

```
for i:=1 to k do
if SolutiePosibila then
   PrelucrareaSolutiei
```

Soluție Posibilă este o funcție booleana care returnează valoarea true dacă elementul satisface condițiile problemei și false în caz contrar.

Prelucrarea Soluției este o procedură care efectuează prelucrarea elementului selectat. De obicei, ăn această procedură soluția este afișată la ecran

Schema generală a algoritmului [2]



Etapele de studiere [2]

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.

Probleme rezolvate

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. [3]

```
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.
```

2. Să se scrie un program care determină toate secvenţele binare de lungime n, fiecare din ele conţinînd nu mai puţin de k cifre de 1. Intrare: numere naturale n, 1<n<20, şi k, k<n, se citesc de la tastatură. Ieşire: fiecare linie a fişierului text OUT.TXT va conţine câte o secvenţă binară distinctă, ce corespunde condiţiilor din enunţul problemei. [2]

```
Program Triere;
const
nmax=20;
type secventa= array[1..nmax] of 0..1;
var b:secventa;
    r,i,n,k:integer;
function numara:integer;
var s,j:integer;
begin s:=0;
for j:=1 to n do s:=s+b[j];
numara:=s;
end;
procedure scrie;
var j: integer;
begin
for j:=1 to n do write (f,b[j]); writeln(f);
end;
procedure urmator (var x:secventa);
var j:integer
```

```
begin j:=n;
while x[j]=1 do
begin x[j]:=0;
j:=j-1;
end;
x[j]:=1;
end;
begin
readln(n,k);
assign(f,'OUT.TXT');
rewrite(f);
for i:=1 to n do
b[i]:=0;
repeat r:= numara;
if r >= k then scrie;
if r < n then
urmator(b);
until r=n;
close(f);
end.
```

Concluzii

1. Prin metoda trierii programele elaborate afișează pe ecran soluțiile găsite după ce este analizat fiecare element.
2. Deși soluțiile problemei sunt de tip ordinal, ele mai pot fi prezentate sub formă de tablou sau mulțime.
3. Nu toate problemele pot fi rezolvate prin această metodă, dar anume cele care implică enumerarea după calcul, alegerea între elemente.

Bibliografie

- https://www.mindmeister.com/689967199/metoda-trierii?fullscreen=1 [1]
- https://www.slideshare.net/BalanVeronica/metoda-trierii1 [2]
- http://caterinamacovenco.blogspot.com/p/tehnici-de-programare.html [3]