**Metoda reluarii**

**Botnari Dragos**

**cl. a-XI-a “D”**

**I.P.L.T. “Spiru Haret”**

**Cuprins**

**1. Informații generale**

**2. Schema generală**

**3. Concluzie**

**4. Bibliografie**

**Avantaje**

*-prezența condițiilor de continuare*

*(timpul cerut de algoritmul respectiv este mai mic în comparație cu metoda trierii )*

**Dezavantaje**

*-aranjare numai în ordinea apariției în memorie*

La dispozitia celor care rezolva probleme cu ajutorul calculatorului exista mai multe metode . Dintre acestea cel mai des utilizate sunt:

- metoda Greedy;

- metoda Divide et impera;

- metoda Branch and Bound;

- metoda Backtracking;

Metoda Backtracking se aplica problemelor în care solutia poate fi reprezentata sub forma unui vector – x = (x1, x2, x3, …xk,… xn) € S, unde S este multimea solutiilor problemei si S = S1 x S2 x… x Sn, si Si sunt multimi finite având s elemente si xi € si , (¥)i = 1..n.

Pentru fiecare problema se dau relatii între componentele vectorului x, care sunt numite conditii interne; solutiile posibile care satisfac conditiile interne se numesc solutii rezultat. Metoda de generare a tuturor solutiilor posibile si apoi de determinare a solutiilor rezultat prin verificarea îndeplinirii conditiilor interne necesita foarte mult timp.

Metoda backtracking evita aceasta generare si este mai eficienta. Elementele vectorului x, primesc pe rând valori în ordinea crescatoare a indicilor, x[k] va primi o valoare numai daca au fost atribuite valori elementelor x1.. x[k-1]. La atribuirea valorii lui x[k] se verifica îndeplinirea unor conditii de continuare referitoare la x1…x[k-1]. Daca aceste conditii nu sunt îndeplinite, la pasul k, acest lucru înseamna ca orice valori i-am atribui lui x[k+1], x[k+1], .. x[n] nu se va ajunge la o solutie rezultat.

Metoda backtracking construieste un vector solutie în mod progresiv începând cu prima componenta a vectorului si mergând spre ultima cu eventuale reveniri asupra atribuirilor anterioare.

Metoda se aplica astfel :

1) se alege prima valoare sin S1 si I se atribuie lui x1 ;

2) se presupun generate elementele x1…x[k-1], cu valori din S1..S[k-1]; pentru generarea lui x[k] se alege primul element din S[k] disponibil si pentru valoarea aleasa se testeaza îndeplinirea conditiilor de continuare.

Pot aparea urmatoarele situatii :

a) x[k] îndeplineste conditiile de continuare. Daca s-a ajuns la solutia finala (k = n) atunci se afiseaza solutia obtinuta. Daca nu s-a ajuns la solutia finala se trece la generarea elementului urmator – x [k-1];

b) x[k] nu îndeplineste conditiile de continuare. Se încearca urmatoarea valoare disponibila din S[k]. Daca nu se gaseste nici o valoare în S[k] care sa îndeplineasca conditiile de continuare, se revine la elementul x[k-1] si se reia algoritmul pentru o noua valoare a acestuia. Algoritmul se încheie când au fost luate in considerare toate elementele lui S1.

Problema

Dintr-un nr. de 6 cursuri optionale un elev trebuie sa aleaga 3. Sa se afiseze toate posibilitatile de alegere precum si nr. lor.

program cursuri;

const n=6;

p=3;

type stiva=array [1..10] of integer;

var st:stiva;

ev,as:boolean;

k:integer;

procedure init(k:integer;var st:stiva);

begin

if k>1 then st[k]:=st[k-1]

else if k=1 then st[k]:=0;

end;

procedure succesor(var as:boolean;var st:stiva;k:integer);

begin

if st[k]<n-p+k then begin st[k]:=st[k]+1;

as:=true;

end

else as:=false;

end;

procedure valid(var ev:boolean;var st:stiva;k:integer)

var i:integer;

begin

ev:=true;

for i:=1 to k-1 do if st[i]=st[k] then ev:=false;

end;

function solutie(k:integer):boolean;

begin

solutie:=(k=p);

end;

procedure tipar;

var i:integer;

begin

for i:=1 to p do write (st[i]);

writeln;

end;

begin;

k:=1;init(k,st);

while k>0 do

begin

repeat

succesor (as,st,k);

if as then valid(ev,st,k);

until (not as) or (as and ev);

if as then

if solutie(k) then tipar

else begin

k:=k+1;

init(k,st)

end;

else k:=k-1;

end;

readln;

end.

**Concluzii**

-*Metoda reluării este mai eficientă ca*

*metoda trierii*

*-Elemente ale fiecărei mulțimi M sunt ordonate conform unui criteriu bine stabilit*

*-Se verifică anumite condiții de continuare în care se stabilesc situațiile în care are sens să trecem la calculul următor*

**Bibliografie**

***-Manualul de informatică pentru clasa a 11-a , editura ”Știința” , 2014 :***

***Capitolul 5. TEHNICI DE ELABORARE A ALGORITMILOR, Tema 5.4. : Metoda reluării***

**http://www.scritub.com/stiinta/informatica/METODA-BACKTRACKING1055131414.php**