[METODA RELUĂRII]

Proiect individual la informatică

A REALIZAT:
Mihalachi Mihaela,
clasa a XI-a “C”,
IPLT “Spiru Haret”

INFORMAȚIE GENERALĂ:

Metoda reluării reprezintă un algoritm, care posedă mai mulți vectori ce formeaza o martice. Ea consta în efectuarea unei căutari în vederea găsirii unei soluții și revenirea la alte soluții posibile.
Aceasta se utilizează la rezolvarea problemelor de căutare, generând toate soluțiile posibile. Se construiește soluția pas cu pas. Dacă se constată că pentru o anumită valoare nu se poate ajunge la soluție, se renunță la acea valoare și se reia căutarea din punctul unde am rămas.

AVANTAJE:

- programele respective sint relativ simple,iar depanarea lor nu necesica teste complicate
- nu se generează toate soluțiile ci doar cele ce sunt convenabile rezolvării problemei date

DEZAVANTAJE:

- ea mai costisitoare metodă din punct de vedere al timpului de execuţie
- depinde de iscusința și cunostințele pe care le posedă programatorul

EXEMPLE DE PROGRAME:

- Program NOTA_PERSOANA;

```
Type Persoana=Record Nume:String[20];Nota:Array[1..4]of  integer; End;
Var f:File of Persoana;
   Perstemp:Persoana;
Procedure Creare;
Begin
   Writeln('Introd.');
   Assign(f,'Test.jo');
   Rewrite(f);
   Repeat
      With PersTemp do begin
         Write('Numele:');Readln(Nume);
         If Nume='' then break;
         Write('Notele:');Readln(Nota[1],Nota[2],Nota[3],Nota[4]);
      end;
      Write(f,PersTemp);
   Until False;
   Close(f);
End;
```

```
Procedure Citire;
Begin
   Writeln('Introd.');
   Assign(f,'Test.jo');
   Reset(f);
   Repeat
      Read(f,PersTemp);
      With PersTemp do begin
         Writeln('Numele:',Nume);
         Writeln('Notele:',Nota[1],Nota[2],Nota[3],Nota[4]);
      end;
   Until Eof(f);
   Close(f);
End;
BEGIN
 Creare;
 Citire;
END.
```

- Program PERMUTARI;

```
var st:array[1..25] of integer;i,n,p:integer;
procedure init;
beginwrite('N='); readln(n);for i:=1 to 25 dost[i]:=0;end;
function valid(p:integer):boolean;
beginvalid:=true;
for i:=1 to p-1 do
if st[i]=st[p] then valid:=false;
end;
procedure tipar(p:integer);
var i:integer; beginfor i:=1 to p dowriteln(st[i],' ');
end;
procedure back(p:integer);
```

```
 begin p:=1;
{plecam de la primul nivel }
st[p]:=0;
{initializam nivelul cu 0}
while p>0 do
{cat timp stiva nu este vida}
 beginif st[p]<n then
{mai exista valori neincercate pe nivelul p}
 beginst[p]:=st[p]+1;
{st[p]<-<o noua valoare din multimea valorilor posibile>}
 if valid(p) thenif p=n then tipar(p)
{solutia este finala}
 else begin p:=p+1;
{trecem la nivelul urmator}
st[p]:=0;
{initializam valoarea de pe nivel cu 0}
end;
end
else
p:=p-1; {pas inapoi}
 end;
end;
begin
init;
back(1);
end.
```

- ```
  Program Backtracking;
  uses crt;
  type TSolutie=array[0..20] of Byte;
  var v:TSolutie;
  solutii:array[1..100] of TSolutie;
  nrsol:integer;
  n:Byte;
  procedure Citire;
  var f:text;
  ```
  ```

```
begin assign(f,'perm.in');
reset(f);
readln(f,n);
Close(f);
end;
procedure Afisare;
var i,k:byte;
begin for k:=1 to nrsol do
begin
for i:=1 to n do
begin
write(solutii[k,i],',');
end;
writeln;
end;
end;
function LungimeData(k:byte):boolean;
begin LungimeData:=k=n;
end;
function final(k:byte):boolean;
begin final:=LungimeData(k);
end;
function Continuare(pozitie:integer; element:Byte):boolean;
var c:byte;
begin Continuare:=true;
for c:=1 to pozitie-1 do
if v[c]=element  then
begin
Continuare:=false;
exit;
end;
end;
procedure back(k:integer);
var element:Byte;
y0:byte;
begin
if final(k-1) then begin inc(nrsol);
solutii[nrsol]:=v;
end;
else
begin for y0:=1 to n do
begin
element:=y0;
if Continuare(k,y0) then begin v[k]:=element;
back(k+1);
end;
end;
end;
```

```
procedure Initializeaza;
begin nrsol:=0;
end;
begin
Citire;
Initializeaza;
back(1);
Afisare;
end.
```

- ```
  Program Regine;
  const NrMaxRegine = 30;
  type Indice = 0 .. NrMaxRegine;
  Solutie = array[Indice] of 0..NrMaxRegine;
  var C: Solutie;
  n: Indice;
  NrSol: word;
  procedure Afisare;
  var i, j: Indice;
  begin inc(NrSol);
  writeln('Solutia nr. ', NrSol);
  for i := 1 to n do
  begin
  for j := 1 to n do
  if j = C[i] then write(' * ') else write(' o ');
  writeln end;
  writeln;
  readln;
  end;
  procedure Plaseaza_Regina(k: Indice); {cand apelam procedura Plaseaza__Regina cu parametrul k am
  plasat deja regine pe liniile 1, 2, ...,k-l}
  var i, j: Indice; ok: boolean;
  begin
  if k-1 = n then {am obtinut o solutie}
   Afisare {prelucrarea solutiei consta in afisare} else {trebuie sa mai plasam regine pe liniile k,k+l,...,n} for i
  := 1 to n do {determin multimea MC a candidatilor pentru pozitia k}
  begin
  ok := true; {verific daca pot plasa regina de pe linia k in coloana i}
  for j := 1 to k-1 do if (C[j]=i) or (abs(C[j]-i)=(k-j)) then ok := false; {regina s-ar gasi pe aceeasi coloana sau
  aceeasi diagonala cu o regina deja plasata}
  if ok then {valoarea i respecta conditiile interne}
  begin
  C[k] := i;{i este un candidat, il extrag imediat}
  Plaseaza_Regina(k+1) ;
  end;
  end;
  end;
  ```
  ```

```
begin {program principal}
write('n= ');
readln(n);
Plaseaza_Regina(1);
end.
```

- ```
  program ecuatie;
  var a,b,c,d:real;
  BEGIN
  write('a=');
  readln(a);
  write('b=');readln(b);
  write('c=');readln(c);
  if a=0 then
  if b=0 then
  if c=0 then
  writeln('Ecuatie nedeterminata, S=R')
  else writeln('Ecuatia nu are solutii.')
  else writeln('Ecuatie de gradul I cu solutia x=',-c/b:6:2)
  else
  begin
  d:=b*b-4*a*c;
  if d>=0 then
  begin
  writeln('x1=',(-b-sqrt(d))/(2*a):6:2);
  writeln('x2=',(-b+sqrt(d))/(2*a):6:2);
  end
  else writeln('Ecuatia nu are solutii reale.');
  end;
  readln;
  END.
  ```

BIBLIOGRAFIE:

https://www.scribd.com/presentation/392484773/Metoda-Reluarii
https://www.scribd.com/document/13396582/Limbajul-Pascal-Metoda-Backtracking-Permutari
http://www.scritub.com/stiinta/informatica/c/Probleme-rezolvate-si-exerciti63855.php
Manual, clasa XI-a, EDITURA Știința
  ```