METODA „DIVIDE ET IMPERA”

(Desparte și stăpânește)

Căutare binară

```
program cautare;
type vector=array[1..100] of integer;
var a:vector;
n,i,li,ls,k,nr:integer;
gasit:boolean;
begin
write(‘n=’);readln(n);
for i:=1 to n do
begin
write(‘a[‘,i,’]=’);
readln(a[i]);
end;
write(‘nr=’);readln(nr);
li:=1; ls:=n; gasit:=false;
repeat
k:=(li+ls) div 2;
if a[k]=nr
then
begin
writeln(‘gasit pe pozitia ‘,k);
gasit:=true;
end
else
if a[k]<nr
then li:=k+1
else ls:=k-1
until (li>ls) or gasit;
if li>ls then writeln(‘negasit’);
readln
end.
```

Sortarea prin interclasare

```
program SortarePrinInterclasare;
const max=10;
var a: array[1..max] of integer;i,n: 1..max;
procedure Interclaseaza(start,mijloc,finis:integer);
var b: array[1..max] of integer; i,j,k:integer;
begin
k:=start ; i:=start;j:=mijloc+1;
while (i<=mijloc) and (j<=finis) do
if a[i]<a[j] then begin b[k]:=a[i]; i:=i+1; k:=k+1 end
else begin b[k]:=a[j]; j:=j+1; k:=k+1 end;
if i<=mijloc then for j:=i to mijloc do begin b[k]:=a[j];
k:=k+1 end
else for i:=j to finis do begin b[k]:=a[i]; k:=k+1 end;
for i:=start to finis do a[i]:=b[i]
end;
procedure SortInterclas(inceput,sfarsit: Integer);
var centru:Integer;
begin
if inceput < sfarsit then
begin
```

```
centru:=(inceput+sfarsit) div 2;

SortInterclas(inceput,centru);

SortInterclas (centru+1, sfarsit );

Interclaseaza (inceput,centru,sfarsit)

end

end;
```

```
begin

Write(‘n=’); readln(n);

for i:=1 to n do begin write(‘a[‘,i,’]=’);
readln(a[i]) end;

SortInterclas(1,n); for i:=1 to n do
write(a[i],’,’); readln

end.
```

Turnurile din Hanoi

```
program TurnurileDinHanoi;

uses crt;

const pauza=10; forma=#219;

varf:array[1..3] of byte=(13,22,22);

procedure HideCursor; assembler; {ascunde cursorul palpaitor, in modul text}

asm MOV AX, $0100; MOV CX, $2607; INT $10 end;

procedure ShowCursor; assembler; {reafiseaza cursorul}

asm MOV AX, $0100; MOV CX, $0506; INT $10 end;

function ColTija(tija:byte):byte;{stabileste coloana unei tije}

begin ColTija:=24*tija-8 end;

procedure MutaDreapta(disc, tija1, tija2: byte);

var i,k:byte;

begin

for i:=ColTija(tija1)-disc to Pred(ColTija(tija2)-disc) do

begin

Delay(Pauza); if KeyPressed then Halt(1);

gotoxy(i,3); for k:=0 to disc do Write(‘ ‘);
gotoxy(i+1,3); for k:=0 to 2*disc do Write(forma);
end
end;
procedure MutaStanga(disc, tija1, tija2 :byte);
var i,k:byte;
begin
for i:=ColTija(tija1)-disc downto succ(ColTija(tija2)-disc) do
begin
Delay(Pauza); if KeyPressed then Halt(1);
gotoxy(i,3); for k:=0 to disc do Write(‘ ‘);
gotoxy(i-1,3); for k:=0 to 2*disc do Write(forma);
end
end;
```

```
procedure Coboara(disc, tija:byte);
var i,k:byte;
begin
for i:=3 to Pred(Varf[tija]-1) do
begin
Delay(Pauza); if KeyPressed then Halt(1);
gotoxy(ColTija(tija)-disc,i);
for k:=0 to disc do Write(‘ ‘);
gotoxy(ColTija(tija)-disc,i+1);
for k:=0 to 2*disc do Write(forma);
end;
Dec(Varf[tija])
end;
```

```
procedure Ridica(disc, tija:byte);
var i,k:byte;
begin
for i:=Varf[tija] downto 4 do
begin
Delay(Pauza); if KeyPressed then Halt(1);
gotoxy(ColTija(tija)-disc,i);
for k:=0 to disc do Write(‘ ‘);
gotoxy(ColTija(tija)-disc,i-1);
for k:=0 to 2*disc do Write(forma);
end;
Inc(Varf[tija])
end;
procedure Muta(disc, tija1, tija2:byte);
begin
```

```
Ridica(disc,tija1);
if (tija1<tija2) then
MutaDreapta(disc,tija1,tija2);
Coboara(disc, tija2)
end;
procedure Han(n, tija1, tija2, tija3:byte);
begin
if (n=1) then Muta(1,tija1,tija2)
else
begin
Han(n-1, tija1, tija3, tija2);
Muta(n, tija1,tija2);
Han(n-1, tija3, tija2,tija1)
end
end;
```

```
procedure Initializari;
var k,disc:byte;
begin
HideCursor; Clrscr;
for disc:=1 to 9 do
begin
gotoxy(ColTija(1)-disc,varf[1]+disc-1);
for k:=0 to 2*disc do
write(forma);
end
end;
begin
Initializari; gotoxy(28,1); writeln(‘- Turnurile din Hanoi -‘);
Han(8,1,2,3); ShowCursor;
readln
end.
```

Maximul unui vector

```
program maxim;
var v:array[1..10] of integer;
n,i:integer;
function max(i,j:integer):integer;
var a,b:integer;
begin
if i=j then max:=v[i]
else begin
a:=max(i, (i+j) div 2);
b:=max((i+j) div 2+1,j);
if a>b then max:=a
else max:=b;
end;
end;
begin
write(’n=’);
readln(n);
for i:=1 to n do read(v[i]);
writeln(maximul este ’,max(1,n));
end.
```

Cel mai mare divizor comun

```
program cmmdc_sir;
const nmax=20;
type indice=1..nmax;
var a:array[indice] of word;
n:indice;
procedure citire;
var i:indice;
begin
readln(n);
for i:=1 to n do read(a[i]);
end;
```

```
function euclid(x,y:word):word;
var r:word;
begin
while y<>0 do
begin
r:=x mod y;
x:=y;
y:=r;
end;
euclid:=x;
end;
```

```
function cmmdc(p,q:indice):word;
var m:indice;
begin
if q-p<=1 then cmmdc:=euclid(a[p],a[q])
else
begin
m:=(p+q) div 2;
cmmdc:=euclid(cmmdc(p,m),cmmdc(m+1,q));
end;
end;
begin
citire;
writeln(’cmmdc=’,cmmdc(1,n));
readln;
end.
```

Bibliografie

- http://www.creeaza.com/referate/informatica/Metoda-de-programare-DIVIDE-ET449.php
- https://informaticacnet.wordpress.com/category/clasa-a-xi-a/metode-divide-et-impera/

Mulțumesc pentru atenție!