

[METODA DESPARTE ȘI STĂPÎNEȘTE]

Proiect individual la informatică

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

Informatii generale:

Metoda Desparte și Stăpânește presupune:

- 1) împărțirea repetată a unei probleme de dimensiuni mari în două sau mai multe subprobleme de același tip, dar de dimensiuni mai mici;
- 2) rezolvarea subproblemelor în mod direct, dacă dimensiunea lor permite aceasta, sau împărțirea lor în alte subprobleme de dimensiuni și mai mici;
- 3) combinarea soluțiilor subproblemelor rezolvate pentru a obține soluția problemei inițiale.

Probleme rezolvate:

1. Se citește un vector cu n componente, numere naturale. Se cere să se tipărească valoarea maximă.

```
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.
```

2. Fie n valori numere naturale $a_1, a_2, a_3, \dots, a_n$. Determinați cel mai mare divizor comun al lor prin metoda Divide Et Impera.

```
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.

```

3. Se citeste de la tastatura un sir de n elemente numere intregi. Sa se gaseasca elementul aflat pe o pozitie data k in sirul ordonat crescator, fara a efectua ordonarea.

```

program pozitia_k;
const nmax=20;
type indice=1..nmax;
var a:array[indice] of integer;
n,k:indice;
procedure citire;
var i:indice;
begin
write('n=');
readln(n);
for i:=1 to n do begin write('a[' ,i,']=');
readln(a[i]);
end;
end;
procedure afisare;
var i:indice;
begin
writeln('vectorul este');
for i:=1 to n do write(a[i]:4);
writeln;
end;
function divide(p,q:indice):indice;
var st,dr:indice;x:integer;
begin
st:=p;
dr:=q;
x:=a[p];
while st<dr do
begin
while (st<dr) and (a[dr]>=x) do dec(dr);a[st]:=a[dr];
while (st<dr) and (a[dr]<=x) do inc(st);a[dr]:=a[st];

```

```

end;
a[st]:=x;
divide:=st;
end;
function qselect(st,dr,k:indice):integer;
var p:indice;
begin
p:=divide(st,dr);
if k=p-st+1 then qselect:=qselect(st,p-1,k)
else qselect:=qselect(p+1,dr,k-(p-st+1));
end;
begin
citire;
write('k=')
;readln(k);
writeln('pozitia ',k,' in vectorul sortat este ',qselect(1,n,k));
afisare;
end.

```

4. Se considera un sir de n numere intregi, ordonat crescator si un numar intreg x. Sa se partitioneze sirul dat in doua subsiruri, astfel incat toate elementele primului sir sa fie mai mici decat x, iar toate elementele celui de-al doilea sir sa fie mai mari decat x.

```

program partitionare;
type vector=array[1..20] of integer;
var v:vector;n,x,i,ref:integer;
function part(p,q):integer;
var mij:integer;
begin
if q<p then part:=p
else begin mij:=(p+q) div 2;
if x=v[mij] then part:=mij else
if x<v[mij] then part:=part(p,mij-1)
else part:=part(mij+1,q);
end;
end;
begin
write('n=');
readln(n);
write('v[1]=');
readln(v[1]);
for i:=2 to n do
repeat
write('v[',i,']=');
readln(v[i]);
until v[i]>=v[i-1];
write('x=');
readln(x);
ref:=part(1,n);
writeln('primul vector');
for i:=1 to ref-1 do write(v[i]:5);
writeln;
writeln('al doilea vector');
for i:=ref to n do write(v[i]:5);
readln;
end.

```

5. Se da o bucata de tabla cu lungimea l si inaltimea h, avand pe suprafata ei n gauri coordonate numere intregi. Se cere sa se decupeze din ea o bucata de arie maxima care nu prezinta gauri. Sunt permise numai taieturi verticale si orizontale.

```

program taieturi;
type vect=array[1..9] of integer;
var l,h,i,n,xf,yf,lf,hf:integer;
xv,yv:vect;
procedure dimp(x,y,l,h:integer;var xf,yf,lf,hf:integer;var xv,yv:vect);
var gasit:boolean;i:integer;
begin
i:=1;
gasit:=false;
while (i<=n) and (not gasit) do
if (xv[i]>x) and (xv[i]<x+1) and
(yv[i]>y) and (yv[i]<y+h)
then gasit:=true
else i:=i+1;
if gasit then
begin
dimp(x,y,xv[i]-x,h,xf,yf,lf,hf,xv,yv);
dimp(xv[i],y,l+x-xv[i],h,xf,yf,lf,hf,xv,yv);
dimp(x,y,l,yv[i]-y,xf,yf,lf,hf,xv,yv);
dimp(x,yv[i],l,h+y-yv[i],xf,yf,lf,hf,xv,yv);
end
else
if (l*h)>(lf*hf) then begin xf:=x;
yf:=y;
lf:=l;
hf:=h;
end;
end;
begin
write('numarul de gauri=');
readln(n);
for i:=1 to n do
begin
write('x[',i,']=');
readln(xv[i]);
write('y[',i,']=');
readln(yv[i]);
end;
write('lungimea dreptunghiului=');
readln(l);
write('inaltimea dreptunghiului=');
readln(h);
lf:=0;
hf:=0;
dimp(0,0,l,h,xf,yf,lf,hf,xv,yv);
writeln 'x=',xf,' y=',yf,' l=',lf,' h=',hf);
end.

```

Concluzii:

Algoritmii de tip Divide et Impera au buna executare in timp, daca se indeplinesc urmatoarele conditii:

- dimensiunile subprogramelor(in care se imparte problema initiala)sunt aproximativ egale("principiul balansarii");
- lipsesc fazele de combinare a solutiilor subproblemelor(cautare binara).

Bibliografie:

<http://www.creeaza.com/referate/informatica/Metoda-de-programare-DIVIDE-ET449.php>

https://prezi.com/_hzd2fwjkmrm/metoda-desparte-si-stapineste/

<https://www.scribd.com/doc/133583568/Tehnici-de-Programare>

Manual, clasa XI-a, EDITURA Știința