PROIECT INDIVIDUAL LA INFORMATICĂ

TEMA: Desparte și stăpânește (divide et impera)

A REALIZAT: Cernatinschi Nicolae, clasa a XI-a ”C”
A VERIFICAT: Maria Guțu
IPLT ‘Spiru Haret’

INFORMAȚIE

Metoda de programare DIVIDE ET IMPERA constă în împărțirea problemei inițiale de dimensiuni [n] în două sau mai multe probleme de dimensiuni reduse.În general se execută împărțirea în două subprobleme de dimensiuni aproximativ egale și anume [n/2].Împărțirea în subprobleme are loc până când dimensiunea acestora devine suficient de mică pentru a fi rezolvate în mod direct (cazul de bază).După rezolvarea celor două subprobleme se execută faza de combinare a rezultatelor în vederea rezolvării întregii probleme.

AVANTAJE

-Timpul execuției relativ mic

-Programele cu această metodă sunt simple de înțeles

-Această tehnică admite o implementare recursivă

-Această metodă stă la baza a mai multe metode de sortarea rapidă

DEZAVANTAJE

-Metoda poate fi aplicată numai când prelucrarea cerută admite divizarea problemei curente în subprobleme

-Metodă complicată de folosit pentru mulți

-Metodă ce necesită mai multe condiții, astfel este folosit destul de rar

Etapele acestei metode sunt:

- descompunerea problemei inițiale în subprobleme independente similare problemei de bază ,de dimensiuni mai mici;
- descompunerea treptată a subproblemelor în alte subprobleme din ce în ce mai simple ,până

când se pot rezolva imediat prin algoritmul simplificat;
- rezolvarea subproblemelor simple;
- combinarea soluțiilor găsite pentru construirea soluțiilor subproblemelor de dimensiuni din ce în ce mai mari;

- combinarea ultimelor soluții determinate
- obținerea soluției problemei inițiale.

EXEMPLE DE PROGRAME

1)Calcularea maximului dintr-un tablou array

```
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)Găsirea celui mai mare divizor comun dintr-un tablou array

Fie n valori numere naturale a1,a2,a3,..,an. Determinati cel mai mare divizor comun al lor prin metoda Divide Et Impera. Se imparte sirul a1,a2,a3,..,an in doua subsiruri a1,a2,a3,..,am,respectiv am+1,am+2,.,an,unde m reprezinta pozitia de mijloc,m=(1+n) div 2.

Cmmdc(a1,a2,..,an)= Cmmdc(a1,a2,..,am), Cmmdc(am+1,am+2,.,an))

```
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)Sortarea rapidă QuickSort

Un tablou V se completeaza cu n elemente numere reale .Sa se ordoneze crescator folosind metoda de sortare rapida .

Solutia problemei se bazeaza pe urmatoarele etape implementate in programul principal:

- se apeleaza procedura “quick” cu limita inferioara li=1 si limita superioara ls=n;
- functia ”poz” realizeaza mutarea elementului v[i] exact pe pozitia ce o va ocupa acesta in vectorul final ordonat; functia ”poz” intoarce (in k) pozitia ocupata de acest element;
- in acest fel ,vectorul V se imparte in doua parti : li …k-1 si k+1…ls;
- pentru fiecare din aceste parti se reapeleaza procedura“quick”,cu limitele modificate corespunzator ;
- in acest fel primul element din fiecare parte va fi pozitionat exact pe pozitia finala ce o va ocupa in vectorul final ordonat (functia “poz”);
- fiecare din cele doua parti va fi astfel impartita in alte doua parti procesul continua pana cand limitele partilor ajung sa se suprapuna, ceea ce indica ca toate elementele vectorului au fost mutate exact pe pozitiile ce le vor ocupa in vectorul final; deci vectorul este ordonat ;
- În acest moment se produc intoarcerile din apelurile recursive si programul isi termina executia .

```
 program quicksort;

 type  vector= array [1..50] of real;

 var v:vector; i,n,k:integer;

function poz(li,ls:integer):integer;

var i,j,modi,modj,m:integer;

man:real;

begin

i:=li; j:=ls;

modi:=0; modj:=-1;

while i<j do

begin

if  v[i]>v[j] then

begin

man:=v[i];
```

```
 v[i]:=v[j];


v[j]:=man;

m:=modi ;

modi:=-modj;

modj:=-m;

end;

i:=i+modi;

 j:=j+modj;

end;

poz:=i;

end;

procedure   quick(li,ls:integer);

begin

if  li<ls then begin

 k::=poz(li,ls);

quick(li,k-1);

quick(k+1,ls);

end;

end;

begin

write('cate elemente are vectorul ?=');readln(n);

for i:=1 to n do

begin

write('tastati elementul ',i,'=');

readln(v[i]);
```

```
end;

quick(1,n);

writeln('vectorul ordonat este :');

for i:=1 to n do writeln(v[i]);

readln;

end.
```

4)Sortare prin interclasare(mergesort)

Tabloul unidimensional V se completeaza cu n numere reale. Sa se ordoneze crescator folosind sortare prin interclasare.

Sortarea prin interclasare se bazeaza pe urmatoarea logica :

vectorul V se imparte prin injumatatiri succesive in vectori din ce in ce mai mici cand se ating vectorii de maxim doua elemente, fiecare dintre acestia se ordoneaza printr-o simpla comparare a elementelor; cate doi astfel de mini- vectori ordonati se interclaseaza succesiv pana se ajunge iar la vectorul V.

```
program mergesort;

type vector=array[1..50] of real ;

var v:vector ;n,i:word;

procedure schimba(li,ls:word;var a:vector);

var man:real;

begin

if a[li]>a[ls] then begin

man:=a[li];

a[li]:=a[ls];

a[ls]:=man;

end;
```

```
end;

procedure interclas(li,m,ls:word;var a:vector);

var b:vector:i,k,p,j:word;

begin

i:=li; j:=m+1; k:=0;

while (i<=m)and(j<=ls) do

begin

inc(k);

if  a[i] <a[j] then begin

b[k]:=a[i];

inc(i);

end

else  begin

b[k]:=a[j];

inc(j);

end;

end;

if i<=m then for p:=i to m do  begin1

inc(k);b[k]:=a[p];

end;

if j<=ls then for p:=j to ls do begin

inc(k)
;b[k]:=a[p];


end;

k:=0;
```

```
for p:=li to ls do begin
inc(k);
a[p]:=b[k];
end;
end;
procedure divi(li,ls:word; var a:vector);
var m:word;
begin
if  (ls-li)<=1 then schimba(li,ls,a);
else begin
m:=(li+ls)div 2;
divi(li,m,a);
divi(m+1,ls,a);
interclas(li,m,ls,a);
end;
end;
begin
write('cate elemente are vectorul?=');readln(n);
for i:=1 to n do
begin
write('tastati elementul',i,'=');
readln(v[i]);
end;
divi(1,n,v);
writeln('vectorul sortat este:');
```

```
for i:=1 to n do writeln(v[i]);

end.
```

5)Problema plierilor

Se considera un vector de lungime n. Definim plierea vectorului prin suparapunerea unei jumatati, numita donoare, peste cealalta jumatate, numita receptoare, cu precizarea ca daca numarul de elemente este impar, elementul din mijloc este eliminat. Prin pliere, elementele subvectorului obtinut vor avea numeroatrea jumatatii recepetoare. Plierea se poate aplica in mod repetat, pana cand se ajunge la un subvector format dintr-un singur element, numit element final.Scrieti un program care sa afiseze toate elementele finale posibile si sa figureze pe ecran pentru fiecare element final o suucesiune de plieri.

```
program plieri;

const nmax=50;

tyepe element=1..nmax;

var n,i:element;

efinal:array[element] of boolean;

m:array[element] of string;

procedure pliaza(p,q:element);

begin

if p=q then efinal [p]:=true

else

begin

if (q-p+1) mod 2=1 then

begin

efinal[(p+q) div 2]:=false;

ls:=(p+q) div 2-1;

end
```

```
else
ls:=(p+q) div 2;
ld:=(p+q) div 2+1;
pliaza(p,ls);
str(ls,ss);
str(ld,sd);
for i:=p to ls do
m[i]:='d'+sd+'  '+m[i];
end;
end;
begin
write(,n=');
readln(n);
for i:=1 to n do m[i]:='  ';
pliaza(1,n);
writeln('elementele finale sunt:' ');
for i:=1 to n do if efinal[i] then begin write (, ' ': ');
writeln(m[i]);
end;
writeln;
end.
```

CONCLUZIE

Metoda desparte și stăpânește este o temă predestinată doar persoanelor ce tind în a se duce în domeniul IT, astfel această temă fiind opțională în curriculum. Din opiniile a mai multor persoane, această temă este una complicată, dar totuși foarte folositoare în multe cazuri, precum realizarea sortării rapide, prin metoda precum quicksort sau bubblesort. Unele din aceste cazuri le-am enumerat mai sus prin exemple de programe.

BIBLIOGRAFIE

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

MANUAL CLASA XI-a EDITURA Știința

http://lectura.bibliotecadigitala.ro/cazacunina/Caietul%20profesorului_de_Informatica_Cazacu_Nina.pdf

https://www.scribd.com/document/130021652/Algoritmi-in-Pascal

https://www.didactic.ro/materiale-didactice/test-divide-et-impera-pascal

http://www.scritub.com/stiinta/informatica/METODA-DIVIDE-ET-IMPERA25186243.php