МИНИСТЕРСТВО ОБРАЗОВАНИЯ и НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Коломенский институт (филиал)**

федерального государственного бюджетного образовательного учреждения

высшего образования

**«Московский политехнический университет»**

Профиль «**Программное обеспечение вычислительной техники и автоматизированных систем**»

**Направление подготовки:**

**«Информатика и вычислительная техника»**

Форма обучения: очная

**лабораторная работа № 6**

по дисциплине «Структуры и алгоритмы обработки данных»

Выполнил студент группы ИВТ-21 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /И.А. Готманов

(подпись) И.О. Фамилия

Проверил преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ И.Н. Филоненко /

(подпись) И.О. Фамилия

Дата:

Коломна 2020

**Лабораторная работа № 6**

**Деревья , как динамические структуры данных .**

***Цель работы:***

Написать и отладить программу исключения элемента из недвоичного сильноветвящегося В-дерева(Procedure UnderFlow). В программе учесть возможность реорганизации структуры В-дерева в результате исключения элемента со страницы.

***Код программы:***

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, Xpman, StdCtrls, Menus, ExtCtrls, ImgList, Spin, Vcl.ComCtrls;

const

size = 200;

n=2;

nn=4;

type ref=^page;

item=record

key:integer;

p:ref;

count:integer;

end;

page=record

m:0..nn;

p0:ref;

e:array[1..nn] of item;

end;

TForm1 = class(TForm)

Button1: TButton;

SpinEdit1: TSpinEdit;

Panel1: TPanel;

Button2: TButton;

TreeView1: TTreeView;

Memo1: TMemo;

procedure FormCreate(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

l: integer;

tree: ref;

h: boolean;

i: integer;

w: integer;

root,q:ref;

x:integer;

u:item;

implementation

{$R \*.dfm}

procedure search(x:integer; a:ref; var h:boolean; var v:item);

var k,l,r:integer;

q:ref;

u:item;

procedure insert;

var i:integer; b:ref;

begin

with a^ do

begin

if m<nn then

begin

m:=m+1;

h:=false;

for i:=m downto r+2 do e[i]:=e[i-1];

e[r+1]:=u;

end else

begin

new(b);

if r<=n then

begin

if r=n then v:=u else

begin

v:=e[n];

for i:=n downto r+2 do e[i]:=e[i-1];

e[r+1]:=u;

end;

for i:=1 to n do b^.e[i]:=a^.e[i+n]

end else

begin

r:=r-n;

v:=e[n+1];

for i:=1 to r-1 do b^.e[i]:=a^.e[i+n+1];

b^.e[r]:=u;

for i:=r+1 to n do b^.e[i]:=a^.e[i+n];

end;

m:=n;

b^.m:=n;

b^.p0:=v.p;

v.p:=b;

end;

end

end;

BEGIN

if a=nil then

begin

h:=true;

with v do

begin

key:=x;

count:=1;

p:=nil;

end

end

else

with a^ do

begin

l:=1;

r:=m;

repeat

k:=(l+r)div 2;

if x<=e[k].key then r:=k-1;

if x>=e[k].key then l:=k+1;

until r<l;

if l-r>1 then

begin

e[k].count:=e[k].count+1;

h:=false;

end else

begin

if r=0 then q:=p0 else q:=e[r].p;

search(x,q,h,u);

if h then insert

end

end

end;

procedure delete(x:integer; a:ref; var h:boolean);

var i,k,l,r:integer; q:ref;

procedure underflow(c,a:ref; s:integer; var h:boolean);

var b:ref; i,k,mb,mc:integer;

begin

mc:=c^.m;{h=true; a^.m=n-1}

if s<mc then

begin

s:=s+1;

b:=c^.e[s].p;

mb:=b^.m;

k:=(mb-n+1)div 2;

a^.e[n]:=c^.e[s]; a^.e[n].p:=b^.p0;

if k>0 then

begin

for i:=1 to k-1 do a^.e[i+n]:=b^.e[i];

c^.e[s]:=b^.e[k];

c^.e[s].p:=b;

b^.p0:=b^.e[k].p;

mb:=mb-k;

for i:=1 to mb do b^.e[i]:=b^.e[i+k];

b^.m:=mb;

a^.m:=n-1+k;

h:=false;

end else

begin

for i:=1 to n do a^.e[i+n]:=b^.e[i];

for i:=s to mc-1 do c^.e[i]:=c^.e[i+1];

a^.m:=nn;

c^.m:=mc-1; {dispose(b)}

h:=c^.m<n;

end

end else

begin

if s=1 then b:=c^.p0 else b:=c^ .e[s-1].p;

mb:=b^.m+1;k:=(mb-n)div 2;

if k>0 then

begin

for i:=n-1 downto 1 do a^.e[i+k]:=a^.e[i];

a^.e[k]:=c^.e[s];

a^.e[k].p:=a^.p0;

mb:=mb-k;

for i:=k-1 downto 1 do a^.e[i]:=b^.e[i+mb];

a^.p0:=b^.e[mb].p;

c^.e[s]:=b^.e[mb];

c^.e[s].p:=a;

b^.m:=mb-1;

a^.m:=n-1+k;

h:=false;

end else

begin

b^.e[mb]:=c^.e[s];

b^.e[mb].p:=a^.p0;

for i:=1 to n-1 do b^.e[i+mb]:=a^.e[i];

b^.m:=nn;

c^.m:=mc-1; {dispose(a)}

h:=c^.m<n

end

end

end;

procedure del(p:ref; var h:boolean);

var q:ref;

begin

with p^ do

begin

q:=e[m].p;

if q<> nil then

begin

del(q,h);

if h then underflow(p,q,m,h)

end else

begin

p^.e[m].p:=a^.e[k].p;

a^.e[k]:=p^.e[m];

m:=m-1;

h:=m<n

end

end

end;

BEGIN

if a=nil then

begin

Form1.Panel1.Caption:='Искомый ключ не найден'; h:=false

end else

with a^ do

begin

l:=1;

r:=m;

repeat

k:=(l+r)div 2;

if x<=e[k].key then r:=k-1;

if x>=e[k].key then l:=k+1;

until l>r;

if r=0 then q:=p0 else q:=e[r].p;

if l-r>1 then

begin

if q=nil then

begin

m:=m-1;

h:=m<n;

for i:=k to m do e[i]:=e[i+1];

end else

begin

del(q,h);

if h then underflow(a,q,r,h);

end

end else

begin

delete(x,q,h);

if h then underflow(a,q,r,h);

end

end

end;

procedure printtree(p:ref; l:integer; treenode:TTreeNode);

var

i:integer;

h:integer;

node: TTreeNode;

node\_val: string;

lvl:integer;

begin

lvl := 0;

node\_val := '';

if p <> nil then

with p^ do

begin

for i:= 1 to l do

lvl := lvl + 1;

Form1.Memo1.Lines.Add('');

Form1.Memo1.Lines.Add('-- ' + inttostr(lvl) + ' --');

Form1.Memo1.Lines.Add('');

for i:=1 to m do

begin

Form1.Memo1.Lines.Add(inttostr(e[i].key));

node\_val := node\_val + inttostr(e[i].key) +' ';

end;

node := Form1.TreeView1.Items.AddChild(treenode, node\_val);

printtree(p0,l+1, node);

for i:=1 to m do printtree(e[i].p,l+1, node);

end;

end;

//----------------------------------------------------------//

procedure TForm1.FormCreate(Sender: TObject);

begin

randomize();

tree:=nil;

h:=false;

end;

procedure TForm1.Button1Click(Sender: TObject);

begin

for i:=1 to 30 do

begin

x:=random(40)+1;

search(x,tree,h,u);

if h then

begin

q:=tree;

new(tree);

with tree^ do

begin

m:=1;

p0:=q;

e[1]:=u;

end;

end;

end;

Button1.Enabled:=false;

SpinEdit1.Visible:=true;

Button2.Visible:=true;

printtree(tree, 1, nil);

//treeout(nil, tree);

end;

procedure TForm1.Button2Click(Sender: TObject);

begin

Panel1.Caption:='';

x:=SpinEdit1.Value;

delete(x,tree,h);

if h then

begin

if tree^.m=0 then

begin

q:=tree;

tree:=q^.p0; {dispose(q)}

end

end;

Form1.TreeView1.Items.Clear;

Form1.Memo1.Clear;

printtree(tree, 1, nil);

end;

end.







