**Двоичный**

**function** divide(L,R:int):int;

var m:word;

begin if R<=L then result:=R

else begin m:=(L+R) div 2;

if arr[m].n<k then

result:=divide(m+1, R)

else result:=divide(l,m);

end; end;

**Слиянием**

**procedure** Tzad.sortsliv;

**procedure** slip(L,m,R:word);

var i,j,k:word; c:mas;

begin i:=l; k:=1; j:=m+1;

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

if a[i].k<a[j].k then

begin c[k]:=a[i]; inc(i); inc(k); end

else begin c[k]:=a[j]; inc(j); inc(k);

end; while i<=m do begin

c[k]:=a[i]; inc(i); inc(k); end;

while j<=R do begin c[k]:=a[j];

inc(j); inc(k); end; k:=0;

for i:=L to R do begin inc(k);

a[i]:=c[k]; end; end;

**procedure** srsl(L,R:word);

var m:word; begin

if l<>r then begin

m:=(l+r) div 2; srsl(l,m);

srsl(m+1,r); slip(l,m,r);

end; end;

Begin srsl(1,n); end;

**Хоара**

**procedure** Tzad.quicksort;

**procedure** sort(L,R:word);

var i,j:word; w:Tinf; X:tkey;

begin i:=L; j:=R;

x:=a[(L+R) div 2].k;

repeat

while a[i].k<x do inc(i);

while a[j].k>x do dec(j);

if i<=j then begin w:=a[i];

a[i]:=a[j]; a[j]:=w; inc(i);

dec(j); end; until j<i;

if L<=j then sort(L,j);

if i<=R then sort(i,R);

end; begin sort(1,n); end;

**Полный перебор**

**procedure** Tzad.PP(i:word);

begin wt:=wt+a[i].w;

ct:=ct+a[i].c; **include**(s,i);

if i<n then PP(i+1)

else if (wt<=wmax) and (ct>cmax) then

begin sopt:=s; cmax:=ct; end;

**exclude**(s,i); wt:=wt-a[i].w;

ct:=ct-a[i].c; if i<n then PP(i+1)

else if (wt<=wmax) and (ct>cmax) then

begin sopt:=s; cmax:=ct; end; end;

**Ветви+границы**

procedure Tzad.VG(i:word;wt,oct:Integer);

var wt1,oct1:integer; begin wt1:=wt+a[i].w;

if wt1<=wmax then begin include(s,i);

if i<n then vg(i+1,wt1,oct) else if

oct>Cmax then begin sopt:=s; cmax:=oct;

end; exclude(s,i); end; oct1:=oct-a[i].c;

if oct1>cmax then if i<n then

vg(i+1,wt,oct1) else begin sopt:=s;

cmax:=oct1; end; end;

**список пузырька**

**constructor** Tzad.create;

begin inherited create;

new(sp1); sp1.a:=nil; end;

**procedure** Tzad.add1(inf:Tinf);

begin new(sp); sp.inf:=inf;

sp.a:=sp1.a; sp1.a:=sp; end;

**procedure** Tzad.read1(var Inf:Tinf);

begin sp:=sp1.a; inf:=sp.inf;

sp1.a:=sp.a; dispose(sp); end;

**procedure** Tzad.sort;

**procedure** revafter(spi:Psel);

var p:Psel; begin p:=spi.a.a;

spi.a.a:=p.a; p.a:=spi.a;

spi.a:=p; end; var spt:Psel;

begin spt:=nil; repeat

sp:=sp1; while sp.a.a<>spt do

begin if sp.a.inf.k>sp.a.a.inf.k then

revafter(sp); sp:=sp.a; end; spt:=sp.a;

until spt=sp1.a.a;end;

**Односвязный слияние**

constructor Tzad.create; begin

inherited create; sp1:=nil;

spk:=nil; end;

**procedure** Tzad.addk(inf:Tinf);

begin new(sp); sp.inf:=inf;

sp.a:=nil; if sp1=nil then

begin sp1:=sp; spk:=sp;

end else begin spk.a:=sp;

spk:=sp; end;end;

**procedure** Tzad.read1(var Inf:Tinf);

begin sp:=sp1; inf:=sp.inf;

sp1:=sp.a; if sp1=nil then

spk:=nil; dispose(sp); end;

**procedure** Tzad.sortslip;

**procedure** div2sp(tp:Tzad; var tq,tr:Tzad);

var c:integer; inf:Tinf; begin

tq:=Tzad.create; tr:=Tzad.create;

c:=-1; while tp.sp1<>nil do

begin c:=-c; tp.read1(inf);

if c>0 then tq.addk(inf) else tr.addk(inf);

end; end; //div2sp

**procedure** slip(tq,tr:Tzad;var tp:Tzad);

var inf:Tinf; begin while (tq.sp1<>nil)

and(tr.sp1<>nil) do if tq.sp1.inf.k>tr.sp1.inf.k

then begin tr.read1(inf); tp.addk(inf); end

else begin tq.read1(inf); tp.addk(inf);

end; while tq.sp1<>nil do begin tq.read1(inf);

tp.addk(inf); end; while tr.sp1<>nil do

begin tr.read1(inf); tp.addk(inf); end; end;

**procedure** srsl(tp:Tzad); var tq,tr:Tzad;

begin if tp.sp1<>tp.spk then begin

div2sp(tp,tq,tr); srsl(tq); srsl(tr); slip(tq,tr,tp);

end; end; //srsl

begin srsl(self) end;

**Решить выражение**

**procedure** Tstack.add1;

begin new(sp); sp.inf:=inf;

sp.a:=sp1; sp1:=sp; end;

**procedure** Tstack.read1;

begin sp:=sp1; inf:=sp.inf;

sp1:=sp.a; dispose(sp);end;

**Дерево**

**procedure** Tzad.Wrt1B(var sgrid:sgrd);

var i:word; **procedure** wrt(p:ptree);

begin if p<>nil then begin wrt(p.a1);

inc(i); sgrid.Cells[0,i]:=p.inf.f;

sgrid.Cells[1,i]:=inttostr(p.inf.k);

wrt(p.a2); end; end; begin i:=0;

p:=proot; wrt(p); end;

**procedure** tzad.AddB(inf:Tinf);

var bl:boolean; begin new(w);

w.inf:=inf; w.a1:=nil; w.a2:=nil;

if proot=nil then proot:=w

else begin p:=proot; repeat

q:=p; bl:=inf.k<p.inf.k; if bl then

p:=p.a1 else p:=p.a2; until p=nil;

if bl then q.a1:=w else q.a2:=w;

end;end;