Міністерство освіти, науки, молоді та спорту України

Львівський національний університет імені Івана Франка

Кафедра радіофізики

Звіт

про виконання лабораторної роботи №2

« Методи сортування цілих чисел:

швидке сортування Хоара, сортування злиттям »

Студентки

Групи ФеІ – 22

Литвин Віри

Перевірив

доц. Вельгош С.Р.

Львів 2012

**Тема :** Методи сортування чисел. Сортування злиттям, швидке сортування Хоара.

**Мета:** вивчити методи швидкого сортування чисел злиттям та Хоара і якнайраціональніше їх практично реалізувати.

**Теоретичний матеріал:**

**Швидке сортування** ([англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *Quick Sort*) — [алгоритм сортування](http://uk.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D1%81%D0%BE%D1%80%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), добре відомий, як алгоритм розроблений [Чарльзом **Хоаром**](http://uk.wikipedia.org/wiki/%D0%A7%D0%B0%D1%80%D0%BB%D1%8C%D0%B7_%D0%A5%D0%BE%D0%B0%D1%80), який не потребує додаткової пам'яті і виконує у середньому \;O(n\log\;n) операцій. Однак, у найгіршому випадку робить *O*(*n*2) порівнянь. Оскільки алгоритм використовує дуже прості цикли і операції, він працює швидше інших алгоритмів, що мають таку ж асимптотичну оцінку складності.

Ідея алгоритму полягає в переставлянні елементів масиву таким чином, щоб його можна було розділити на дві частини і кожний елемент з першої частини був не більший за будь-який елемент з другої. Впорядкування кожної з частин відбувається рекурсивно. Алгоритм швидкого сортування може бути реалізований як у [масиві](http://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%81%D0%B8%D0%B2_(%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85)), так і в двозв’язному [списку](http://uk.wikipedia.org/wiki/%D0%97%D0%B2%27%D1%8F%D0%B7%D0%B0%D0%BD%D0%B8%D0%B9_%D1%81%D0%BF%D0%B8%D1%81%D0%BE%D0%BA).

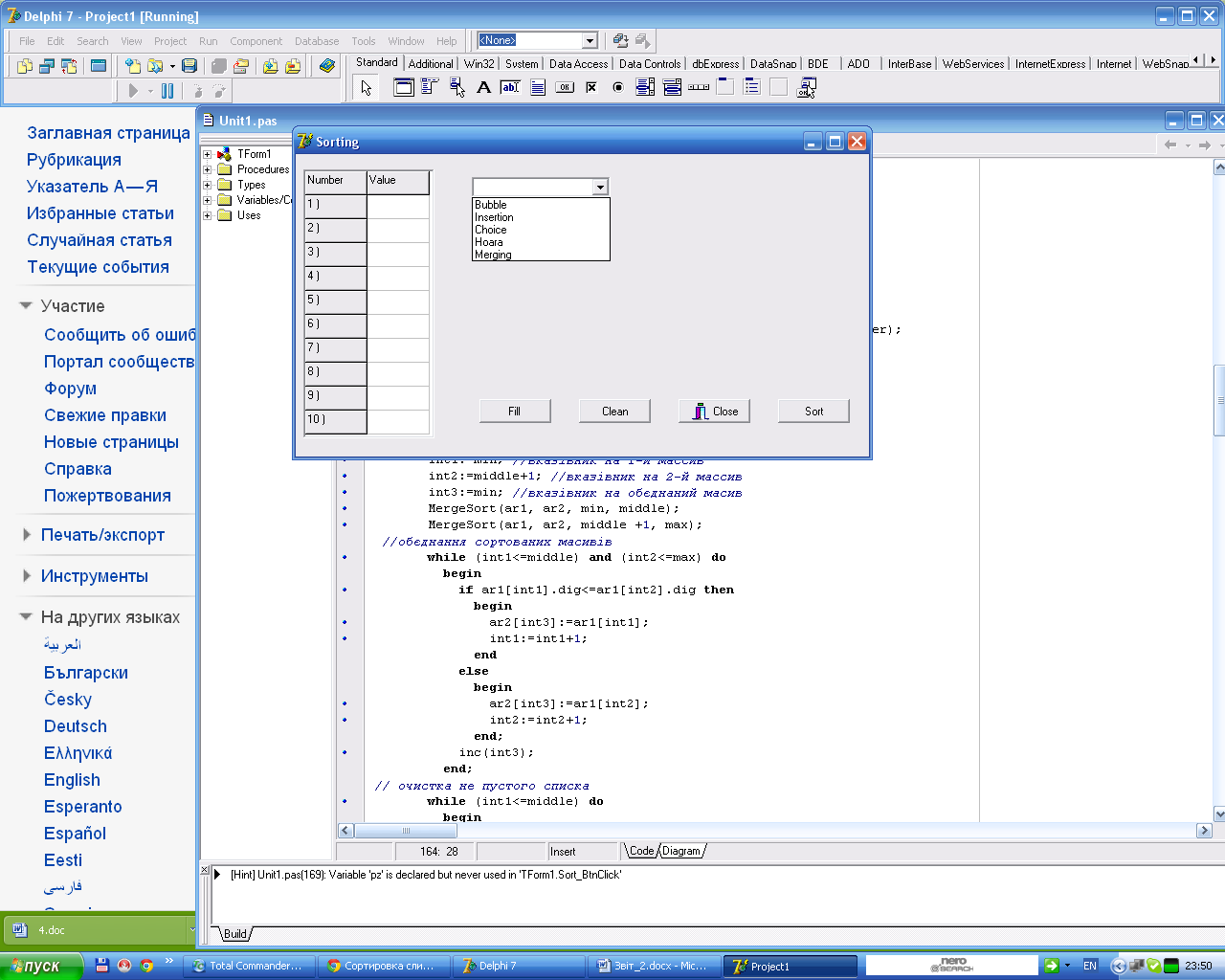
Швидке сортування є [алгоритмом на основі порівнянь](http://uk.wikipedia.org/w/index.php?title=%D0%A1%D0%BE%D1%80%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%BD%D0%B0_%D0%BE%D1%81%D0%BD%D0%BE%D0%B2%D1%96_%D0%BF%D0%BE%D1%80%D1%96%D0%B2%D0%BD%D1%8F%D0%BD%D1%8C&action=edit&redlink=1), і не є [стабільним](http://uk.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%B1%D1%96%D0%BB%D1%8C%D0%BD%D0%B5_%D1%81%D0%BE%D1%80%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F).

**Сортування злиттям** — алгоритм сортування, в основі якого лежить принцип “[Розділяй та володарюй](http://uk.wikipedia.org/w/index.php?title=%D0%A0%D0%BE%D0%B7%D0%B4%D1%96%D0%BB%D1%8F%D0%B9_%D1%82%D0%B0_%D0%B2%D0%BE%D0%BB%D0%BE%D0%B4%D0%B0%D1%80%D1%8E%D0%B9&action=edit&redlink=1)”. В основі цього способу сортування лежить злиття двох упорядкованих ділянок масиву в одну впорядковану ділянку іншого масиву.

Під час сортування в дві допоміжні черги з основної поміщаються перші дві відсортовані підпослідовності, які потім зливаються в одну і результат записується в тимчасову чергу. Потім з основної черги беруться наступні дві відсортовані підпослідовності і так до тих пір доки основна черга не стане порожньою. Після цього послідовність з тимчасової черги переміщається в основну чергу. І знову продовжується сортування злиттям двох відсортованих підпослідовностей. Сортування триватиме до тих пір поки довжина відсортованої підпослідовності не стане рівною довжині самої послідовності.

**Практична реалізація:**

Для програмної реалізації я використала середовище Deiphi 7. На формі створила наступний інтерфейс:

Тут дані не потрібно вводити, вони заповнюються автоматично та нумеруються, за допомогою оператора Random, при натисканні кнопки **Fill.**

Кнопка **Clean** витирає дані з колонки Value значення і впорядковує нумерацію Number.

При натисканні Close програма закриється.

А Sort відсортовує числа, переставляючи їх разом з номером.

Проте, аби сортувати дані, спершу потрібно обрати один із методів сортування: Hoara(швидке сортування), Merging(злиттям).

**Висновок**

Швидке сортування є [алгоритмом на основі порівнянь](http://uk.wikipedia.org/w/index.php?title=%D0%A1%D0%BE%D1%80%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%BD%D0%B0_%D0%BE%D1%81%D0%BD%D0%BE%D0%B2%D1%96_%D0%BF%D0%BE%D1%80%D1%96%D0%B2%D0%BD%D1%8F%D0%BD%D1%8C&action=edit&redlink=1), і не є [стабільним](http://uk.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%B1%D1%96%D0%BB%D1%8C%D0%BD%D0%B5_%D1%81%D0%BE%D1%80%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F" \o "Стабільне сортування). Алгоритм сортування злиттям не міняє порядок розташування однакових елементів, а отже він є [стабільним](http://uk.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%B1%D1%96%D0%BB%D1%8C%D0%BD%D0%B5_%D1%81%D0%BE%D1%80%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F" \o "Стабільне сортування).

Обоє алгоритмів виконують приблизно однакову кількість операцій ,але швидке сортування не потребує додаткової пам’яті.

Одже, алгоритм потрібно обирати для певного класу задач.

**Додаток. Код програми:**

**Швидке сортування методом Хоара.**

begin

L := Ryad[1];

R := Ryad[10];

Metod\_Hoara(L, R);

for i:=1 to n do

begin

StringGrid1.Cells[0, i]:=IntToStr(Ryad[i].n)+' )';

StringGrid1.Cells[1, i]:=IntToStr(Ryad[i].dig);

end;

End;

procedure Metod\_Hoara(L, R: Point);

var i, j, Change: Point;

pz : Integer;

Begin

i := L;

j := R;

pz := Ryad[(L.n+R.n) div 2].dig;

repeat

while Ryad[i.n].dig < pz do

inc(i.n);

while Ryad[j.n].dig > pz do

dec(j.n);

if i.n <= j.n then

begin

Change := Ryad[i.n];

Ryad[i.n] := Ryad[j.n];

Ryad[j.n] := Change;

inc (i.n);

dec (j.n);

end;

until i.n >= j.n;

if L.n < j.n then

Metod\_Hoara(L, j);

if i.n < R.n then

Metod\_Hoara(i, R);

End;

**Сортування злиттям.**

begin

begin

L := Ryad[1];

R := Ryad[10];

for i := 1 to 6 do

begin

Sorted[i].dig := 0;

Sorted[i].n := i;

end;

SortMerge(Ryad, Length(Ryad), Sorted);

for i:=1 to n do

begin

StringGrid1.Cells[0, i]:=IntToStr(Ryad[i].n)+' )';

StringGrid1.Cells[1, i]:=IntToStr(Ryad[i].dig);

end;

end;

end;

procedure SortMerge(var Mas: array of Point; MasLength: Integer; var MasTemp: array of Point);

var i, j, n1, n2, N, N0, s, p1, p2: integer;

a: Point;

begin

if MasLength>1 then begin

N:=trunc(0.5\*MasLength);

for i:=1 to N do

if Mas[2\*i-2].dig > Mas[2\*i-1].dig then

begin a:=Mas[2\*i-2]; Mas[2\*i-2]:=Mas[2\*i-1]; Mas[2\*i-1]:=a; end;

s:=2;

N:=Ceil(0.5\*MasLength);

repeat

N0:=N div 2;

for i:=1 to N0 do

begin

p1:=2\*s\*(i-1);

n1:=s;

p2:=p1 + n1;

n2:=s;

if (i=N0)and(2\*N0=N)then n2:=MasLength - (2\*N0-1)\*s;

for j:=0 to n2-1 do MasTemp[j]:=mas[p2+j];

p2:=0;

repeat

if (n1>0)and(n2>0) then

if mas[p1+n1-1].dig >= masTemp[p2+n2-1].dig then

begin Mas[p1 + n1+n2-1]:=mas[p1+n1-1]; n1:=n1-1; end

else

begin Mas[p1 + n1+n2-1]:=masTemp[p2+n2-1]; n2:=n2-1; end;

if (n1=0)and(n2>0) then

begin

for j:=0 to n2-1 do Mas[2\*s\*(i-1) + j]:=masTemp[p2+j];

n2:=0;

end;

if (n1>0)and(n2=0) then n1:=0;

until n1+n2=0;

end;

N:=Ceil(0.5\*N);

s:=s\*2;

until N=1;

end;

end;