Nama : Irfan Satrio Nugroho
NPM : 140810180003

Tugas 4

Merge Sort

```
/*
     Nama         : Irfan Satrio Nugroho
     Kelas        : A
     NPM          : 140810180003
     Nama Program : Mengurutkan elemen menggunakan merge sort
*/

#include <iostream>
#include <chrono>
using namespace std;

void satu(int* in, int p, int q,int r){
    int n1 = q-p+1;
    int n2 = r-q;
    int L[n1+1];
    int R[n2+1];
    for (int i=1; i<=n1; i++){
        L[i-1] = in[(p-1)+i-1];
    }

    for (int j=1; j<=n2; j++){
        R[j-1] = in[(q-1)+j];
    }

    int i=0;
    int j=0;
    L[n1]=2147483647;
    R[n2]=2147483647;

    for (int k=(p-1); k<r; k++){
        if(L[i]<=R[j]){
            in[k]=L[i];
            i = i+1;
        }
        else{
            in[k]=R[j];
            j = j+1;
        }
    }
}

void msort(int* in, int p, int r){
    int q;
    if(p<r){
        q = (p+r)/2;
        msort(in, p, q);
        msort(in, q+1, r);

        satu(in, p, q, r);
    }
}

void input(int* a, int& n){
    cout << "Input banyak data: "; cin >> n;
    for (int i=0; i<n; i++){
        cout << "Input angka: "; cin >> a[i];
    }
}

int main(){
    int in[100];
    int n;
    input(in,n);
    auto start = chrono::steady_clock::now();
    msort(in,1,n);
    auto end = chrono::steady_clock::now();
    cout << "Hasil: ";
    for(int i=0; i<n; i++){
        cout << in[i] << " ";
    }

    cout<<endl;
    cout << "Elapsed time in nanoseconds : "
           << chrono::duration_cast<chrono::nanoseconds>(end -
start).count()
           << " ns" << endl;

    return 0;
}
```

Kompleksitas Algoritma merge sort adalah O(n lg n). Cari tahu kecepatan komputer Anda dalam memproses program. Hitung berapa running time yang dibutuhkan apabila input untuk merge sort-nya adalah 20?

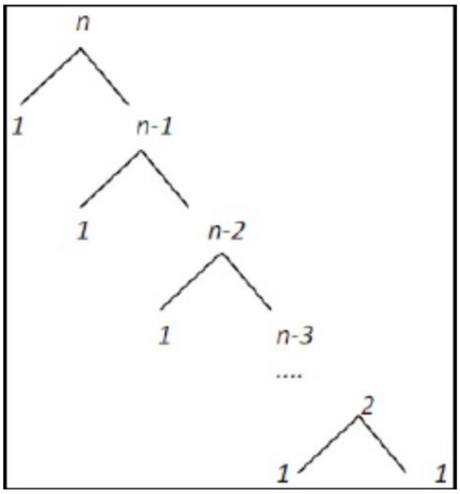
Untuk di program hasilnya : 2369 ns
Tapi jika sesuai dengan O -> $T(20\ \log_{10} 20) = 26$

Selection Sort

```
for i ← n downto 2 do {pass sebanyak n-1 kali}
    imaks ← 1
    for j ← 2 to i do
      if $x_j > x_{imaks}$ then
        imaks ← j
      endif
    endfor
    {pertukarkan $x_{imaks}$ dengan $x_i$}
    temp ← $x_i$
    $x_i$ ← $x_{imaks}$
    $x_{imaks}$ ← temp
endfor
```

Subproblem = 1
Masalah setiap subproblem = n-1
Waktu proses pembagian = n
Waktu proses penggabungan = n$T(n) = \{\Theta(1)\ T(n-1) + \Theta(n)$

T(n) = cn + cn-c +cn-2c + ... + 2c +cn
= c((n-1)(n-2)/2) + cn
= c((n^2-3n+2)/2) + cn
= c(n^2/2)-(3n/2)+1 + cn
=O(n^2)



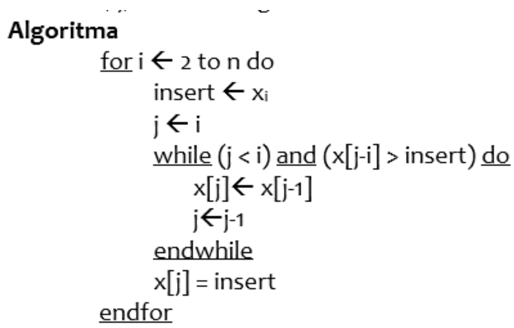
T(n) = cn + cn-c +cn-2c + ... + 2c +cn
= c((n-1)(n-2)/2) + cn
= c((n^2-3n+2)/2) + cn
= c(n^2/2)-(3n/2)+1 + cn
= Ω (n^2)

T(n) = cn^2
= Θ(n^2)

Source Code :

```
/*
	Nama		: Irfan Satrio Nugroho
	Kelas		: A
	NPM		: 140810180003
	Nama Program : Selection Sort
*/
#include <iostream>
#include<conio.h>

using namespace std;

int data[100],data2[100];
int n;

void tukar(int a, int b)
{
	int t;
	t = data[b];
	data[b] = data[a];
	data[a] = t;
}
void selection_sort()
{
     int pos,i,j;
     for(i=1;i<=n-1;i++)
     {
         pos = i;
         for(j = i+1;j<=n;j++)
         {
             if(data[j] < data[pos]) pos = j;
           }
        if(pos != i) tukar(pos,i);
    }
}

int main()
{
     cout << "\n======================================";
     cout<<"\nMasukkan Jumlah Data : ";cin>>n;
     cout << "\n--------------------------------------" << endl;
     for(int i=1;i<=n;i++)
     {
          cout<<"Masukkan data ke-"<<i<<" : ";
          cin>>data[i];
          data2[i]=data[i];
     }

     selection_sort();
     cout << "\n--------------------------------------" << endl;
     cout<<"Data Setelah di Sort : "<<endl;
     for(int i=1; i<=n; i++)
     {
          cout<<" "<<data[i];
     }

     cout << "\n======================================\n";
     getch();
}
```

Insertion Sort



```
Algoritma
     for i ← 2 to n do
          insert ← $x_i$
          j ← i
          while (j < i) and (x[j-i] > insert) do
               x[j]← x[j-1]
               j←j-1
          endwhile
          x[j] = insert
     endfor
```

Subproblem = 1

Masalah setiap subproblem = n-1

Waktu proses penggabungan = n

Waktu proses pembagian = n

$T(n) = \{\Theta(1)\ T(n-1) + \Theta(n)$

$T(n) = cn + cn\text{-}c + cn\text{-}2c + ... + 2c + cn <= 2cn^2 + cn^2$
$\quad = c((n-1)(n-2)/2) + cn <= 2cn^2 + cn^2$
$\quad = c((n^2-3n+2)/2) + cn <= 2cn^2 + cn^2$
$\quad = c(n^2/2)-c(3n/2)+c+cn <= 2cn^2 + cn^2$
$\quad = O(n^2)$

$T(n) = cn <= cn$
$\quad = \Omega\ (n)$

$T(n) = (cn + cn^2)/n$
$\quad = \Theta(n)$

Source Code :

```
/*
      Nama         : Irfan Satrio Nugroho
      Kelas        : A
      NPM          : 140810180003
      Nama Program : Insertion sort
*/

#include <iostream>
#include <conio.h>

using namespace std;

int data[100],data2[100],n;

void insertion_sort()
{
      int temp,i,j;
      for(i=1;i<=n;i++){
          temp = data[i];
            j = i -1;
          while(data[j]>temp && j>=0){
                 data[j+1] = data[j];
               j--;
          }
          data[j+1] = temp;
      }
}
int main()
{
      cout << "\n====================================="<<endl;
      cout<<"Masukkan Jumlah Data : "; cin>>n;
      cout<<endl;
      cout << "\n-------------------------------------" << endl;
      for(int i=1;i<=n;i++)
      {
        cout<<"Masukkan data ke "<<i<<" : ";
        cin>>data[i];
```

```
          data2[i]=data[i];
        }
        cout << "\n----------------------------------" << endl;
        insertion_sort();
        cout<<"\nHasil Sort "<<endl;
        for(int i=1; i<=n; i++)
        {
          cout<<data[i]<<" ";
        }
        cout << "\n=================================="<<endl;
        getch();
}
```

Bubble Sort

Subproblem = 1
Masalah setiap subproblem = n-1
Waktu proses pembagian = n
Waktu proses penggabungan = n

$$T(n) = \{\Theta(1)\ T(n-1) + \Theta(n)$$

$T(n) = cn + cn-c +cn-2c + ... + 2c +c <= 2cn^2 + cn^2$
$= c((n-1)(n-2)/2) + c<= 2cn^2 + cn^2$
$= c((n^2-3n+2)/2) + c<= 2cn^2 + cn^2$
$= c(n^2/2)-c(3n/2)+2c <= 2cn^2 + cn^2$
$=O(n^2)$

$T(n) = cn + cn-c +cn-2c + ... + 2c +c <= 2cn^2 + cn^2$
$= c((n-1)(n-2)/2) + c<= 2cn^2 + cn^2$
$= c((n^2-3n+2)/2) + c<= 2cn^2 + cn^2$
$= c(n^2/2)-c(3n/2)+2c <= 2cn^2 + cn^2$
$= \Omega\ (n^2)$

$T(n) = cn^2 + cn^2$
$= \Theta(n^2)$

Source Code :

```
/*
        Nama         : Irfan Satrio Nugroho
        Kelas        : A
        NPM          : 140810180003
        Nama Program : Bubble Sort
*/

#include <iostream>
#include <conio.h>

using namespace std;


int main(){
        int arr[100],n,temp;
        cout                                                   <<
"\n================================================"<<endl;
```

```
     cout<<"Massukan banyak elemen : ";cin>>n;
     cout  <<  "\n------------------------------------------------"  <<
endl;

     for(int i=0;i<n;++i){
          cout<<"Masukkan Elemen ke "<<i+1<<" : ";cin>>arr[i];
     }

     for(int i=1;i<n;i++){
          for(int j=0;j<(n-1);j++){
               if(arr[j]>arr[j+1]){
                    temp=arr[j];
                    arr[j]=arr[j+1];
                    arr[j+1]=temp;
               }
          }
     }
     cout  <<  "------------------------------------------------"  <<
endl;
     cout<<"\nHasil : "<<endl;
     for(int i=0;i<n;i++){
          cout<<" "<<arr[i];
     }
     cout                                                           <<
"\n================================================"<<endl;
}
```