

Alvina Vania Kirana
140810180010
Worksheet 4

Kasus 1-Merge Sort

```
/*
Alvina Vania Kirana
140810180010
Praktikum Analisis Algoritma
Worksheet 4 kasus 1
Merge sort
*/
```

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

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

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

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

```
    for (int k=(a-1); k<c; 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 a, int c){
    int b;
    if(a<c){
        b = (a+c)/2;
        msort(in, a, b);
        msort(in, b+1, c);

        satu(in, a, b, c);
    }
}

void input(int* x, int& n){
    cout<<"MERGE SORT"<<endl;
    cout << "Masukkan banyak data: "; cin >> n;
    for (int i=0; i<n; i++){
        cout<<"Masukkan data ke-"<

```

```

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

return 0;
}

```

```

Last login: Tue Mar 24 19:08:18 on ttys000
/Users/alvinavania/Desktop/analgo/tugas4/StudiKasus1
alvinas-MacBook-Pro:~ alvinavania$ /Users/alvinavania/Desktop/
iKasus1
MERGE SORT
Masukkan banyak data: 20
Masukkan data ke-0 : 77
Masukkan data ke-1 : 80
Masukkan data ke-2 : 1
Masukkan data ke-3 : 11
Masukkan data ke-4 : 45
Masukkan data ke-5 : 85
Masukkan data ke-6 : 66
Masukkan data ke-7 : 43
Masukkan data ke-8 : 29
Masukkan data ke-9 : 13
Masukkan data ke-10 : 09
Masukkan data ke-11 : 30
Masukkan data ke-12 : 31
Masukkan data ke-13 : 54
Masukkan data ke-14 : 14
Masukkan data ke-15 : 12
Masukkan data ke-16 : 1
Masukkan data ke-17 : 3
Masukkan data ke-18 : 5
Masukkan data ke-19 : 9
Hasil: 1 1 3 5 9 9 11 12 13 14 29 30 31 43 45 54 66 77 80 85
Elapsed time in nanoseconds : 4538 ns
alvinas-MacBook-Pro:~ alvinavania$ 

```

Kompleksitas waktu 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?

Running time yang dibutuhkan untuk menginput 20 inputan adalah 4538 ns

Kasus 2-Selection Sort

```

/*
Alvina Vania Kirana
140810180010
Praktikum Analisis Algoritma
Worksheet 4 kasus 2
Selection sort
*/

```

```

#include <iostream>
using namespace std;

```

```
int x[50],x2[50];
int n;

void tukar(int a, int b)
{
    int t;
    t = x[b];
    x[b] = x[a];
    x[a] = t;
}

void selectionSort()
{
    int pos,i,j;
    for(i=1; i<=n-1; i++)
    {
        pos = i;
        for(j=i+1; j<=n; j++)
        {
            if(x[j] < x[pos]) pos=j;
        }
        if(pos != i) tukar(pos,i);
    }
}

int main()
{
    cout<< "Selection Sort"<<endl;
    cout<<"Masukkan Jumlah Data : ";cin>>n;
    for(int i=1;i<=n;i++)
    {
        cout<<"Masukkan data ke-<<i<<" : ";
        cin>>x[i];
        x2[i]=x[i];
    }

    selectionSort();
    cout<<"Hasil Sorting: "<<endl;
    for(int i=1; i<=n; i++)
    {
```

```

        cout<<" "<<x[i];
    }
    cout<<endl;
}

/Users/alvinavania/Desktop/analgo/tugas
alvinas-MacBook-Pro:~ alvinavania$ /Use
iKasus2
Selection Sort
Masukkan Jumlah Data : 5
Masukkan data ke-1 : 56
Masukkan data ke-2 : 77
Masukkan data ke-3 : 13
Masukkan data ke-4 : 22
Masukkan data ke-5 : 1
Hasil Sorting:
1 13 22 56 77
alvinas-MacBook-Pro:~ alvinavania$ 

```

Tentukan $T(n)$ dari rekurensi (pengulangan) selection sort berdasarkan penentuan rekurensi divide & conquer:

$$T(n) = \begin{cases} \Theta(1) & \text{if } n \leq c \\ aT\left(\frac{n}{b}\right) + D(n) + C(n) & \text{otherwise} \end{cases}$$

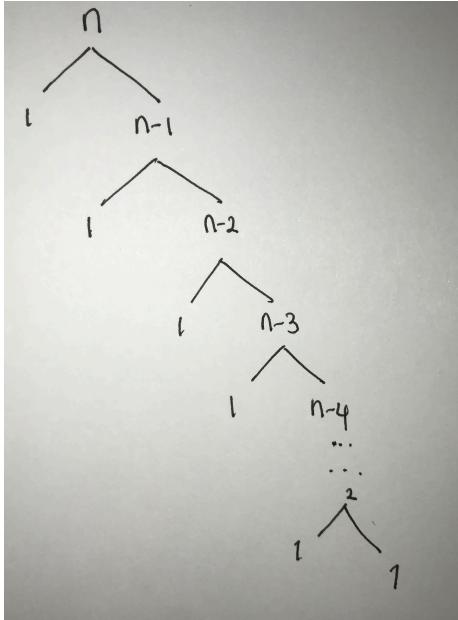
Selesaikan persamaan rekurensi $T(n)$ dengan metode recursion-tree untuk mendapatkan kompleksitas waktu asimptotiknya dalam Big-O, Big- Ω , dan Big- Θ

```

for i ← n downto 2 do {pass sebanyak n-1 kali}
    imaks ← 1
    for j ← 2 to i do
        if xj > ximaks then
            imaks ← j
        endif
    endfor
    {pertukarkan ximaks dengan xi}
    temp ← xi
    xi ← ximaks
    ximaks ← temp
endfor

```

Subproblem	1
Masalah setiap subproblem	n-1
Waktu pembagian	n
Waktu pembagian	n



$$\begin{aligned}
 T(n) &= Cn + cn - c + cn - 2c + \dots + 2c + cn \\
 &= C \left(\frac{(n-1)(n-2)}{2} \right) + cn \\
 &= C \left(\frac{n^2 - 3n + 2}{2} \right) + cn \\
 &= C \left(\frac{n^2}{2} - \frac{3n}{2} \right) + 1 + cn = O(n^2)
 \end{aligned}
 \quad \text{Big-O}$$

Big Θ
 $T(n) = Cn + cn - c + cn - 2c + \dots + 2c + cn$
 $= C \left(\frac{(n-1)(n-2)}{2} \right) + cn$
 $= C \left(\frac{n^2 - 3n + 2}{2} \right) + cn$
 $= C \left(\frac{n^2}{2} - \frac{3n}{2} \right) + cn = \underline{\underline{O}(n^2)}$

Big Ω
 $\text{Big } O = \text{Big } \Omega \rightarrow \underline{\underline{O}(n^2)}$.

Kasus 3-Insertion Sort

```
/*
Alvina Vania Kirana
140810180010
Praktikum Analisis Algoritma
Worksheet 4 kasus 3
Insertion sort
*/
```

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

```
int x[100], x2[100], n;
```

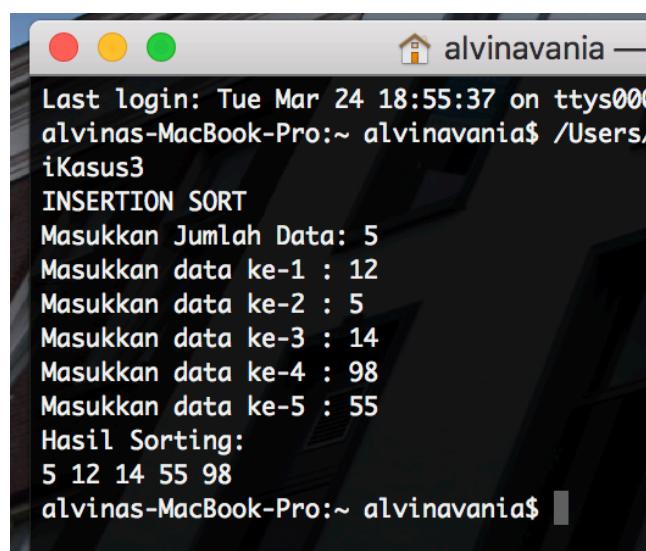
```
void insertionSort()
{
    int temp, i, j;
    for(i=1; i<=n; i++){
        temp=x[i];
        j = i-1;
        while(x[j]>temp && j>=0){

```

```

        x[j+1] = x[j];
        j--;
    }
    x[j+1] = temp;
}
int main()
{
    cout<< "INSERTION SORT"<<endl;
    cout<<"Masukkan Jumlah Data: ";
    cin>>n;
    for(int i=1;i<=n;i++)
    {
        cout<<"Masukkan data ke-"<<i<<" : ";
        cin>>x[i];
        x2[i]=x[i];
    }
    insertionSort();
    cout<<"Hasil Sorting: "<<endl;
    for(int i=1; i<=n; i++)
    {
        cout<<x[i]<<" ";
    }
    cout<<endl;
}

```



Tentukan $T(n)$ dari rekurensi (pengulangan) insertion sort berdasarkan penentuan rekurensi divide & conquer:

$$T(n) = \begin{cases} \Theta(1) & \text{if } n \leq c \\ aT\left(\frac{n}{b}\right) + D(n) + C(n) & \text{otherwise} \end{cases}$$

Selesaikan persamaan rekurensi $T(n)$ dengan metode substitusi untuk mendapatkan kompleksitas waktu asimptotiknya dalam Big-O, Big-Ω, dan Big-Θ

Big O

$$\begin{aligned} T(n) &= cn + cn - c + cn - 2c + \dots + 2c + cn \leftarrow 2n^2 + cn^2 \\ &= C \left(\frac{(n-1)(n-2)}{2} \right) + cn \quad \leftarrow 2cn^2 + cn^2 \\ &= C \left(\frac{n^2 - 3n + 2}{2} \right) + cn \quad \leftarrow 2cn^2 + cn^2 \\ &\approx C \left(\frac{n^2}{2} - \frac{3n}{2} \right) + cn \approx \mathcal{O}(n^2) \quad \leftarrow 2cn^2 + cn^2 \end{aligned}$$

Big Ω

$$T(n) = cn = \Omega(n) \quad \leftarrow cn.$$

Big Θ

$$T(n) = \frac{(cn + cn^2)}{n} = \Theta(n) \quad \leftarrow cn^2 + cn.$$

Kasus 4-Bubble Sort

```
/*
Alvina Vania Kirana
140810180010
Praktikum Analisis Algoritma
Worksheet 4 kasus 4
Bubble sort
*/
```

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

int main()
{
    cout<<"Bubble Sort"<<endl;
    int n, x[100], i ,j ,temp;
    cout<<"Masukkan Jumlah Data: ";cin>>n;
    for(i=0; i<n; i++){
        for(j=i+1; j<n; j++){
            if(x[i] > x[j]){
                temp = x[i];
                x[i] = x[j];
                x[j] = temp;
            }
        }
    }
    cout<<"Hasil Pengurutan: ";
    for(i=0; i<n; i++){
        cout<<x[i]<<" ";
    }
}
```

```

        cout<<"Masukan angka ke-"<<i+1<<": ";
        cin>>x[i];
    }

```

```

for(i=0; i<(n-1); i++) {
    for(j=0; j<(n-i-1); j++)
    {
        if(x[j]>x[j+1])
        {
            temp=x[j];
            x[j]=x[j+1];
            x[j+1]=temp;
        }
    }
}

```

```

cout<<"Hasil Sorting:"<<endl;
for(i=0; i<n; i++){
    cout<<x[i]<<" ";
}
cout<<endl;
}

```

```

$ /Users/alvinavania/Desktop/andigo/tugas4/iKasus4
alvinas-MacBook-Pro:~ alvinavania$ /Users,
iKasus4
Bubble Sort
Masukkan Jumlah Data: 4
Masukan angka ke-1: 77
Masukan angka ke-2: 21
Masukan angka ke-3: 60
Masukan angka ke-4: 1
Hasil Sorting:
1 21 60 77
alvinas-MacBook-Pro:~ alvinavania$ 

```

Tentukan $T(n)$ dari rekurensi (pengulangan) bubble sort berdasarkan penentuan rekurensi divide & conquer:

$$T(n) = \begin{cases} \Theta(1) & \text{if } n \leq c \\ aT\left(\frac{n}{b}\right) + D(n) + C(n) & \text{otherwise} \end{cases}$$

Selesaikan persamaan rekurensi $T(n)$ dengan metode master untuk mendapatkan kompleksitas waktu asimptotiknya dalam Big-O, Big- Ω , dan Big- Θ

Big O

$$\begin{aligned} T(n) &= cn + cn - c + cn - 2c + \dots + 2c + c && \leftarrow 2cn^2 + cn^2 \\ &\geq C \left(\frac{(n-1)(n-2)}{2} \right) + c && \leftarrow 2cn^2 + cn^2 \\ &\geq C \left(\frac{n^2 - 3n + 2}{2} \right) + c && \leftarrow 2cn^2 + cn^2 \\ &\geq C \left(\frac{n^2}{2} - \frac{3n}{2} \right) + 2c = O(n^2) && \leftarrow 2cn^2 + cn^2 \end{aligned}$$

Big Ω

$$\begin{aligned} T(n) &= cn + cn - c + cn - 2c + \dots + 2c + c && \leftarrow 2cn^2 + cn^2 \\ &\leq C \left(\frac{(n-1)(n-2)}{2} \right) + c && \leftarrow 2cn^2 + cn^2 \\ &\leq C \left(\frac{n^2 - 3n + 2}{2} \right) + c && \leftarrow 2cn^2 + cn^2 \\ &\leq C \left(\frac{n^2}{2} - \frac{3n}{2} \right) + 2c = \Omega(n^2) && \leftarrow 2cn^2 + cn^2 \end{aligned}$$

Big Θ

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