

NAMA : AITHRA JUNIA BOUTY

NPM : 140810180054

TUGAS 4

Studi kasus 1: MERGE SORT

1. Source code:

```
#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;
}

```

}

```
Input banyak data: 20
Input angka: 9
Input angka: 8
Input angka: 7
Input angka: 6
Input angka: 5
Input angka: 4
Input angka: 3
Input angka: 2
Input angka: 1
Input angka: 24
Input angka: 35
Input angka: 74
Input angka: 11
Input angka: 348
Input angka: 34
Input angka: 98
Input angka: 90
Input angka: 13
Input angka: 22
Input angka: 51
Hasil: 1 2 3 4 5 6 7 8 9 11 13 22 24 34 35 51 74 90 98 348
Elapsed time in nanoseconds : 4490 ns
```

2. Untuk di program hasilnya : 4490 ns
Tapi jika sesuai dengan O $\rightarrow T(20 \log_{10} 20) = 26$

Studi Kasus 2: SELECTION SORT

- $T(n)$ selection sort
for $i \leftarrow n$ downto 2 do {pass sebanyak $n-1$ kali}
 $imax \leftarrow 1$
 for $j \leftarrow 2$ to i do
 if $x_j > x_{imax}$ then
 $imax \leftarrow j$
 endif
 endfor
 {pertukaran x_{imax} dengan x_i }
 $temp \leftarrow x_i$
 $x_i \leftarrow x_{imax}$
 $x_{imax} \leftarrow temp$
endfor

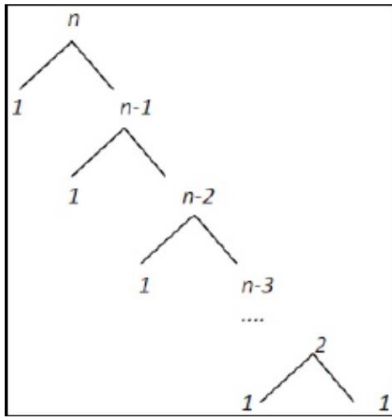
Subproblem = 1

Masalah setiap subproblem = $n-1$

Waktu proses pembagian = n

Waktu proses penggabungan = n

$$T(n) = \{\theta(1) T(n-1) + \theta(n)\}$$



- $$\begin{aligned}
 T(n) &= cn + cn - c + cn - 2c + \dots + 2c + cn \\
 &= c((n-1)(n-2)/2) + cn \\
 &= c((n^2 - 3n + 2)/2) + cn \\
 &= c(n^2/2) - (3n/2) + 1 + cn \\
 &= O(n^2)
 \end{aligned}$$

$$\begin{aligned}
 T(n) &= cn + cn - c + cn - 2c + \dots + 2c + cn \\
 &= c((n-1)(n-2)/2) + cn \\
 &= c((n^2 - 3n + 2)/2) + cn \\
 &= c(n^2/2) - (3n/2) + 1 + cn \\
 &= \Omega(n^2)
 \end{aligned}$$

$$\begin{aligned}
 T(n) &= cn^2 \\
 &= \Theta(n^2)
 \end{aligned}$$

- Koding

```

#include<iostream>
using namespace std;

void swapping(int &a, int &b) {
    int temp;
    temp = a;
    a = b;
    b = temp;
}

void display(int *array, int size) {
    for(int i = 0; i < size; i++)
        cout << array[i] << " ";
    cout << endl;
}

```

```

void selectionSort(int *array, int size) {
    int i, j, imin;
    for(i = 0; i<size-1; i++) {
        imin = i;
        for(j = i+1; j<size; j++)
            if(array[j] < array[imin])
                imin = j;
        swap(array[i], array[imin]);
    }
}

int main() {
    int n;
    cout << "Enter the number of elements: ";
    cin >> n;

    int arr[n];
    cout << "Enter elements:" << endl;
    for(int i = 0; i<n; i++) {
        cin >> arr[i];
    }

    cout << "Array before Sorting: ";
    display(arr, n);
    selectionSort(arr, n);
    cout << "Array after Sorting: ";
    display(arr, n);
}

```

```

"D:\Documents\SEMESTER 4\Analgo\praktikum\Analgo4\selection.exe"
Enter the number of elements: 5
Enter elements:
76
1
32
888
9
Array before Sorting: 76 1 32 888 9
Array after Sorting: 1 9 32 76 888

Process returned 0 (0x0)   execution time : 6.539 s
Press any key to continue.

```

Studi Kasus 3: INSERTION SORT

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

Subproblem = 1

Masalah setiap subproblem = $n-1$

Waktu proses pembagian = n

Waktu proses penggabungan = n

$$T(n) = \{\theta(1) T(n-1) + \theta(n)\}$$

$$\begin{aligned} T(n) &= cn + cn - c + cn - 2c + \dots + 2c + cn \leq 2cn^2 + cn^2 \\ &= c((n-1)(n-2)/2) + cn \leq 2cn^2 + cn^2 \\ &= c((n^2 - 3n + 2)/2) + cn \leq 2cn^2 + cn^2 \\ &= c(n^2/2) - c(3n/2) + c + cn \leq 2cn^2 + cn^2 \\ &= O(n^2) \end{aligned}$$

$$T(n) = cn \leq cn$$

$$= \Omega(n)$$

$$T(n) = (cn + cn^2)/n$$

$$= \Theta(n)$$

- Program

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

void display(int *array, int size) {
    for(int i = 0; i<size; i++)
        cout << array[i] << " ";
    cout << endl;
}

void insertionSort(int *array, int size) {
```

```

    int key, j;
    for(int i = 1; i<size; i++) {
        key = array[i];
        j = i;
        while(j > 0 && array[j-1]>key) {
            array[j] = array[j-1];
            j--;
        }
        array[j] = key;
    }
}

int main() {
    int n;
    cout << "Enter the number of elements: ";
    cin >> n;

    int arr[n];
    cout << "Enter elements:" << endl;

    for(int i = 0; i<n; i++) {
        cin >> arr[i];
    }

    cout << "Array before Sorting: ";
    display(arr, n);
    insertionSort(arr, n);
    cout << "Array after Sorting: ";
    display(arr, n);
}

```

```

"D:\Documents\SEMESTER 4\Analgo\praktikum\Analgoku4\insertion.exe"
Enter the number of elements: 5
Enter elements:
67
3
54
111
2
Array before Sorting: 67 3 54 111 2
Array after Sorting: 2 3 54 67 111

Process returned 0 (0x0)   execution time : 5.340 s
Press any key to continue.

```

Studi Kasus 4: BUBBLE SORT

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

$$T(n) = \{\theta(1) T(n-1) + \theta(n)\}$$

$$\begin{aligned} T(n) &= cn + cn - c + cn - 2c + \dots + 2c + c \leq 2cn^2 + cn^2 \\ &= c((n-1)(n-2)/2) + c \leq 2cn^2 + cn^2 \\ &= c((n^2 - 3n + 2)/2) + c \leq 2cn^2 + cn^2 \\ &= c(n^2/2) - c(3n/2) + 2c \leq 2cn^2 + cn^2 \\ &= O(n^2) \end{aligned}$$

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

$$\begin{aligned} T(n) &= cn^2 + cn^2 \\ &= \Theta(n^2) \end{aligned}$$

- Program

```
#include<iostream>

using namespace std;

int main()
{
    int a[50],n,i,j,temp;
    cout<<"Enter the size of array: ";
    cin>>n;

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

    for(i=1;i<n;++i)
    {
```



```

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

    cout<<"Array after bubble sort:";
    for(i=0;i<n;++i)
        cout<<" "<<a[i];

    return 0;
}

```

"D:\Documents\SEMESTER 4\Analgo\praktikum\Analgoku4\bubble.exe"

```

Enter the size of array: 5
Elemen 1 : 65
Elemen 2 : 45
Elemen 3 : 88
Elemen 4 : 1
Elemen 5 : 3
Array after bubble sort: 1 3 45 65 88
Process returned 0 (0x0)   execution time : 8.141 s
Press any key to continue.

```