Nama : Muhammad Fahmi Alwan

NPM : 140810170052

Kelas : B

Judul : Laprak 4

**Kasus 1 (Merge Sort)**

* **Program Merge Sort**

#include <iostream>

#include <chrono>

using namespace std;

void Merge(int \*a, int low, int high, int mid)

{

int i, j, k, temp[high-low+1];

i = low;

k = 0;

j = mid + 1;

while (i <= mid && j <= high)

{

if (a[i] < a[j])

{

temp[k] = a[i];

k++;

i++;

}

else

{

temp[k] = a[j];

k++;

j++;

}

}

while (i <= mid)

{

temp[k] = a[i];

k++;

i++;

}

while (j <= high)

{

temp[k] = a[j];

k++;

j++;

}

for (i = low; i <= high; i++)

{

a[i] = temp[i-low];

}

}

void MergeSort(int \*a, int low, int high)

{

int mid;

if (low < high)

{

mid=(low+high)/2;

MergeSort(a, low, mid);

MergeSort(a, mid+1, high);

Merge(a, low, high, mid);

}

}

int main()

{

int n, i;

cout<<"-----PROGRAM MERGE SORT-----"<<endl;

cout<<"\nMasukan jumlah data elemen yang ingin diurutkan: ";

cin>>n;

int arr[n];

for(i = 0; i < n; i++)

{

cout<<"Masukan elemen "<<i+1<<": ";

cin>>arr[i];

}

auto start = chrono::steady\_clock::now();

MergeSort(arr, 0, n-1);

//cetak

cout<<"\nHasil Sorting : ";

for (i = 0; i < n; i++)

cout<<"->"<<arr[i];

auto end = chrono::steady\_clock::now();

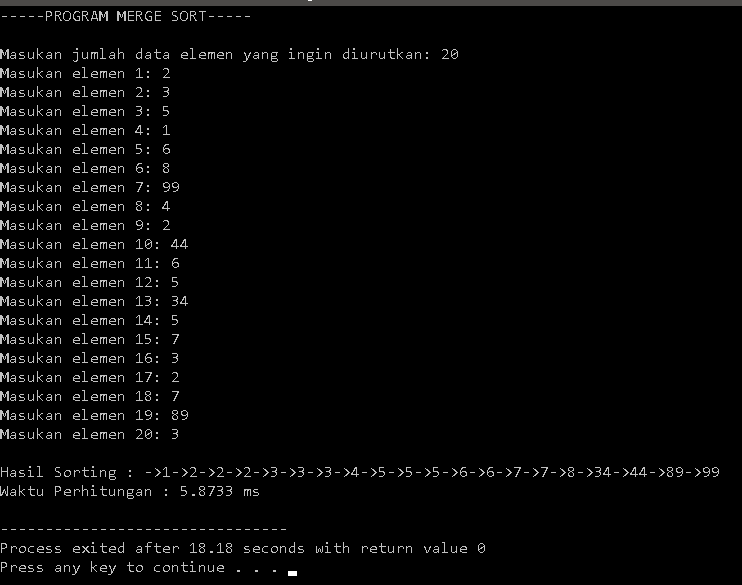
auto diff = end - start;

cout << "\nWaktu Perhitungan : " <<chrono::duration <double, milli> (diff).count() << " ms" << endl;

return 0;

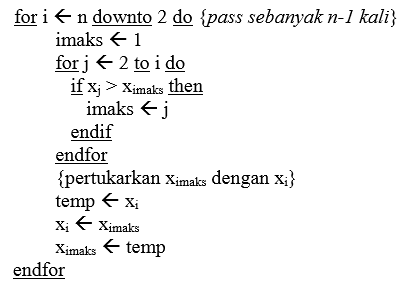
}

* **Hasil Input 20 data**



**Kasus 2 (Selection Sort)**

* **Pelajari Algoritma**



Subproblem = 1

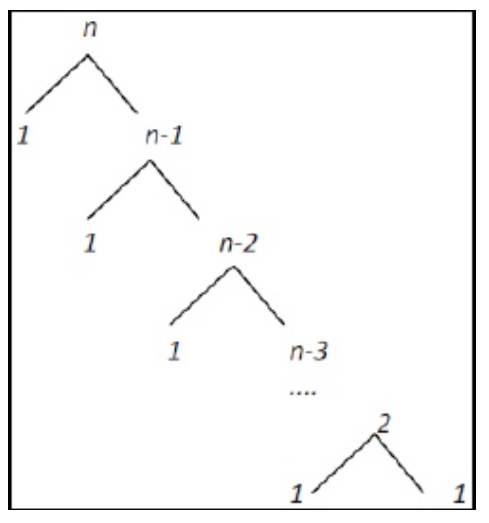
Masalah setiap subproblem = n-1

Waktu proses pembagian = n

Waktu proses penggabungan = n

* **Waktu T(n)**

* **Recursi T(n) dengan recursion Tree**



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)

* **Program**

/\* Nama : Muhammad Fahmi Alwan

NPM : 140810170052

Kelas : B

Judul : Selection Sort

\*/

#include<iostream>

#include<conio.h>

using namespace std;

int main()

{

system("cls");

int size, arr[50], i, j, temp;

cout<<"Ukuran Array : ";

cin>>size;

cout<<"Element Array : ";

for(i=0; i<size; i++)

{

cin>>arr[i];

}

cout<<"Sorting array \n";

for(i=0; i<size; i++)

{

for(j=i+1; j<size; j++)

{

if(arr[i]>arr[j])

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

cout<<"Array setelah sorting :\n";

for(i=0; i<size; i++)

{

cout<<arr[i]<<" ";

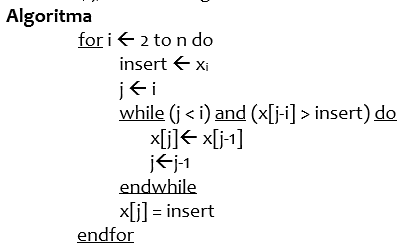
}

getch();

}

**Kasus 3 (Insertion Sort)**

* **Pelajari Algoritma**



Subproblem = 1

Masalah setiap subproblem = n-1

Waktu proses penggabungan = n

Waktu proses pembagian = n

* **Waktu T(n)**

* **Rekursi T(n) Menggunakan Substitusi**

T(n) = cn + cn-c +cn-2c + ..... + 2c +cn <= 2cn^2 + cn^2

= c((n-1)(n-2)/2) + cn<= 2cn^2 + cn^2

= c((n^2-3n+2)/2) + cn<= 2cn^2 + cn^2

= c((n^2)/2)-c(3n/2)+c+cn <= 2cn^2 + cn^2

=O(n^2)

T(n) = cn <= cn

= Ω (n)

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

= Θ(n)

* **Program**

/\* Nama : Muhammad Fahmi Alwan

NPM : 140810170052

Kelas : B

Judul : Insertion Sort

\*/

#include<iostream>

#include<conio.h>

using namespace std;

int main()

{

system("cls");

int size, arr[50], i, j, temp;

cout<<"Ukuran Array : ";

cin>>size;

cout<<"Element Array : ";

for(i=0; i<size; i++)

{

cin>>arr[i];

}

cout<<"Sorting array\n";

for(i=1; i<size; i++)

{

temp=arr[i];

j=i-1;

while((temp<arr[j]) && (j>=0))

{

arr[j+1]=arr[j];

j=j-1;

}

arr[j+1]=temp;

}

cout<<"Array setelah sorting : \n";

for(i=0; i<size; i++)

{

cout<<arr[i]<<" ";

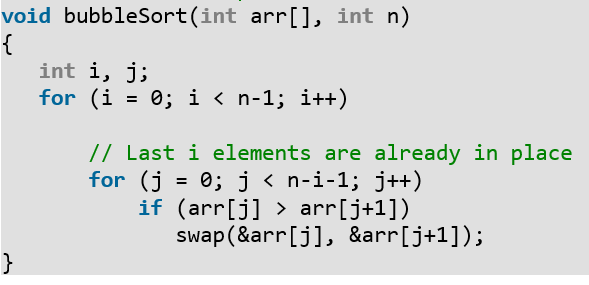
}

getch();

}

**Kasus 4 (Bubble Sort)**

* **Pelajari Algoritma**



Subproblem = 1

Masalah setiap subproblem = n-1

Waktu proses pembagian = n

Waktu proses penggabungan = n

* **Waktu T(n)**
* **Rekursi T(n) dengan Metode Master**

Tidak bisa menggunakan metode master karena B<1 sehingga menggunakan substitusi

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

= Ω (n^2)

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

= Θ(n^2)

* **Program**

#include <iostream>

#include <chrono>

using namespace std;

using namespace std::chrono;

void bubbleSort (int arr[], int n){

int i, j;

for (i = 0; i < n; ++i){

for (j = 0; j < n-i-1; ++j){

if (arr[j] > arr[j+1]){

arr[j] = arr[j]+arr[j+1];

arr[j+1] = arr[j]-arr[j + 1];

arr[j] = arr[j]-arr[j + 1];

}

}

}

}

int main(){

int n, i;

high\_resolution\_clock::time\_point t1 = high\_resolution\_clock::now();

cout<<"Masukkan jumlah elemen data yang ingin diurutkan: ";

cin>>n;

int arr[n];

for(i = 0; i < n; i++){

cout<<"Masukkan elemen ke-"<<i+1<<": ";

cin>>arr[i];

}

bubbleSort(arr, n);

cout<<"\nArray yang telah diurutkan: ";

for (i = 0; i < n; i++){

cout<<" "<<arr[i];

}

high\_resolution\_clock::time\_point t2 = high\_resolution\_clock::now();

auto duration = duration\_cast<microseconds>( t2 - t1 ).count();

cout<<endl<<duration<<" microseconds"<<endl;

}