Arief Ramadhan  
140810170036  
Kelas B

Tugas 4 Praktikum Analisis Algoritma

Kasus 1:

#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 : "

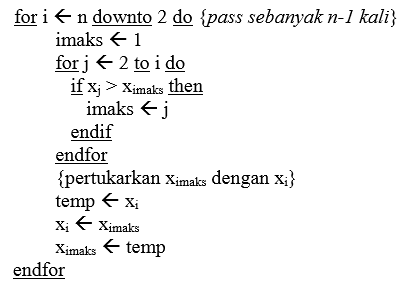
<< chrono::duration\_cast<chrono::nanoseconds>(end - start).count()

<< " ns" << endl;

return 0;

}

Kasus 2:

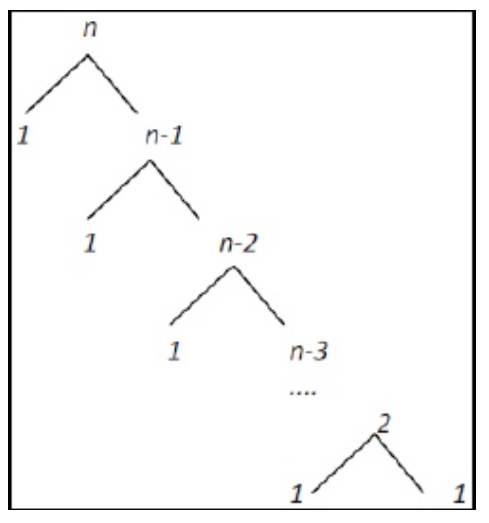


Subproblem = 1

Masalah setiap subproblem = n-1

Waktu proses pembagian = n

Waktu proses penggabungan = 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:

#include <iostream>

#include <conio.h>

using namespace std;

struct masukan{

int in;

masukan\* next;

masukan\* previous;

};

masukan\* input(){

int x;

masukan\* in=NULL;

masukan\* test=NULL;

cout<<"Input banyak data: ";cin>>x;

for (int i=0; i<x; i++){

if(in==NULL){

in = new masukan;

cout<<"Input angka: ";cin>>in->in;

in->next=NULL;

in->previous=NULL;

test=in;

continue;

}

else if(test->next==NULL){

test->next=new masukan;

cout<<"Input angka: ";cin>>test->next->in;

test->next->previous=test;

test->next->next=NULL;

}

test=test->next;

}

return in;

}

void urut(masukan\*& in){

masukan\* test1=in;

masukan\* test2;

masukan\* x;

while(test1->next!=NULL){

test1=test1->next;

}

while(test1!=NULL){

x=in;

test2=in->next;

while(test2!=test1->next){

if(test2->in>x->in){

x=test2;

}

test2=test2->next;

}

swap(test1->in,x->in);

test1=test1->previous;

}

}

int main(){

masukan\* in;

masukan\* sort;

in=input();

urut(in);

masukan\* test=in;

cout<<"Data yang sudah terurut: ";

while(test!=NULL){

cout<<test->in<<" ";

test=test->next;

}

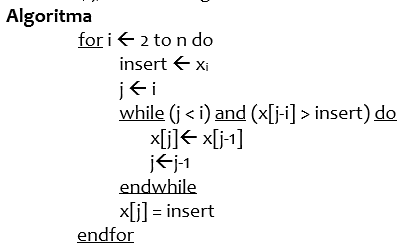
cout<<"\n";

getch();

return 0;

}

Kasus 3:



Subproblem = 1

Masalah setiap subproblem = n-1

Waktu proses penggabungan = n

Waktu proses pembagian = n

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)

Source Code:

#include <iostream>

#include <conio.h>

using namespace std;

struct masukan{

int in;

masukan\* next;

masukan\* previous;

};

masukan\* input(){

int x;

masukan\* in=NULL;

masukan\* test=NULL;

cout<<"Input banyak data: ";cin>>x;

for (int i=0; i<x; i++){

if(in==NULL){

in = new masukan;

cout<<"Input angka: ";cin>>in->in;

in->next=NULL;

in->previous=NULL;

test=in;

continue;

}

else if(test->next==NULL){

test->next=new masukan;

cout<<"Input angka: ";cin>>test->next->in;

test->next->previous=test;

test->next->next=NULL;

}

test=test->next;

}

return in;

}

void urut(masukan\*& in){

masukan\* test1=in;

masukan\* test2;

while(test1->next!=NULL){

test2=test1->next;

while(test2->previous!=NULL && test2->in<test2->previous->in){

swap(test2->in,test2->previous->in);

test2=test2->previous;

}

test1=test1->next;

}

}

int main(){

masukan\* in;

masukan\* sort;

in=input();

urut(in);

masukan\* test=in;

cout<<"Data yang sudah terurut: ";

while(test!=NULL){

cout<<test->in<<" ";

test=test->next;

}

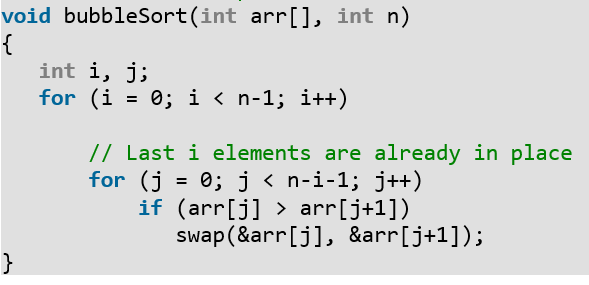
cout<<"\n";

getch();

return 0;

}

Kasus 4:



Subproblem = 1

Masalah setiap subproblem = n-1

Waktu proses pembagian = n

Waktu proses penggabungan = 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

= Ω (n^2)

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

= Θ(n^2)

Source Code:

// Optimized implementation of Bubble sort

#include <stdio.h>

void swap(int \*xp, int \*yp)

{

int temp = \*xp;

\*xp = \*yp;

\*yp = temp;

}

// An optimized version of Bubble Sort

void bubbleSort(int arr[], int n)

{

int i, j;

bool swapped;

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

{

swapped = false;

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

{

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

{

swap(&arr[j], &arr[j+1]);

swapped = true;

}

}

// IF no two elements were swapped by inner loop, then break

if (swapped == false)

break;

}

}

/\* Function to print an array \*/

void printArray(int arr[], int size)

{

int i;

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

printf("%d ", arr[i]);

printf("n");

}

// Driver program to test above functions

int main()

{

int arr[] = {64, 34, 25, 12, 22, 11, 90};

int n = sizeof(arr)/sizeof(arr[0]);

bubbleSort(arr, n);

printf("Sorted array: \n");

printArray(arr, n);

return 0;

}