TUGAS ANALISIS ALGORITMA

Perbandingan Running Time



Oleh:

**Mochamad Faizin A140810160032**

Sachi Hongo 140810160014

Kevin Waworuntu 140810160012

Program Studi S-1 Teknik Informatika

Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Padjadjaran

2019

Binary Search

Code :

#include <iostream>

#include <chrono>

using namespace std;

using namespace std::chrono;

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

void binarySearch(){

const int Ar[10] = {1,2,3,4,5,6,7,8,9,10}; // untuk proses ascending

int tar;

cout<<"masukan data yang dicari : ";

cin>>tar;

int awal=0, akhir=10, tengah;

while (awal <= akhir)

{ tengah = (awal + akhir)/2;

if (tar > Ar[tengah] ) // descending ubah tanda > menjadi <

{ awal = tengah + 1; }

else if (tar < Ar[tengah]) // descending ubah tanda < menjadi >

{akhir= tengah - 1;}

else {awal = akhir +1;

}

}

if (tar == Ar[tengah])

{cout<<" Data ditemukan, Ke- "<<tengah+1<<endl;

}

else {

cout<<"target tidak ditemukan "<<endl;

}

}

int main(int argc, char\*\* argv) {

auto start = high\_resolution\_clock::now();

binarySearch();

auto stop = high\_resolution\_clock::now();

auto duration = duration\_cast<microseconds>(stop-start);

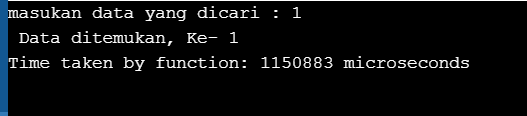
cout<<"Time taken by function: "

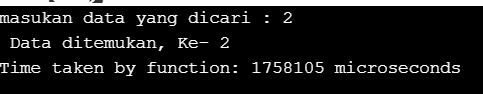
<< duration.count()<<" microseconds"<<endl;

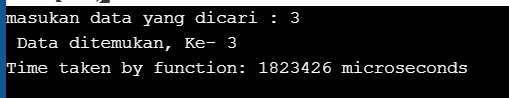
return 0;

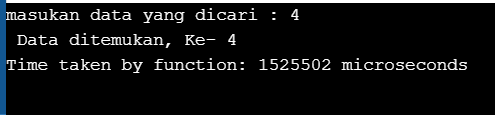
}

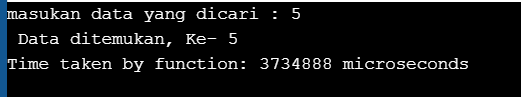
Hasil :

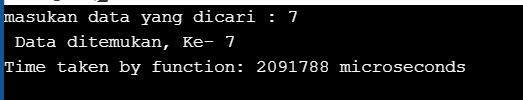
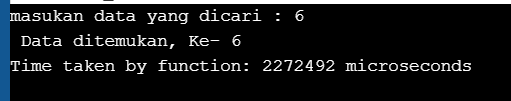


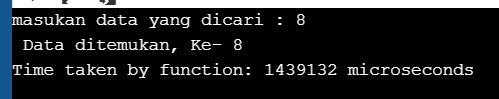


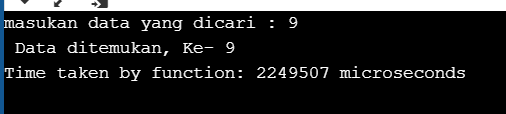


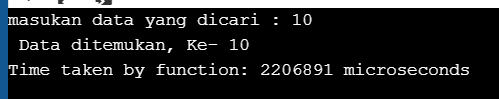


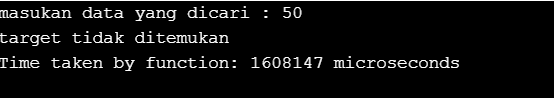
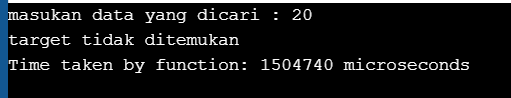












Dari hasil tersebut didapatkan data untuk kecepatan program untuk mencari data berdasarkan algoritma binary search. Namun data waktu yang didapatkan beragam.

Linear Seach:

/\* C++ Program - Linear Search \*/

#include <iostream>

#include <conio.h>

#include <chrono>

using namespace std;

using namespace std::chrono;

void cekLS(){

//clrscr();

int arr[10]= {1,2,3,4,5,6,7,8,9,10};

int i, num;

int c=0, cek;

cout<<"Enter the number to be search : ";

cin>>num;

for(i=0; i<10; i++)

{

if(arr[i]==num)

{

c=1;

cek=i+1;

break;

}

}

if(c==0)

{

cout<<"Number not found..!!";

}

else

{

cout<<num<<" found at position "<<cek;

}

getch();

}

main()

{

auto start = high\_resolution\_clock::now();

cekLS();

auto stop = high\_resolution\_clock::now();

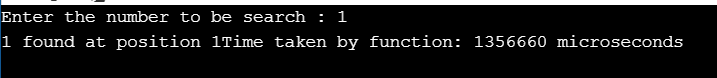
auto duration = duration\_cast<microseconds>(stop-start);

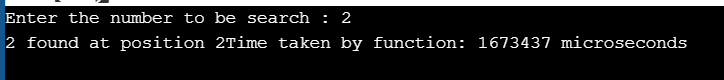
cout<<"Time taken by function: "

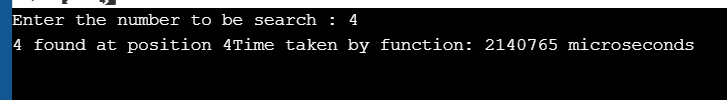
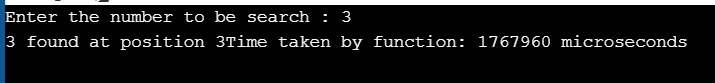
<< duration.count()<<" microseconds"<<endl;

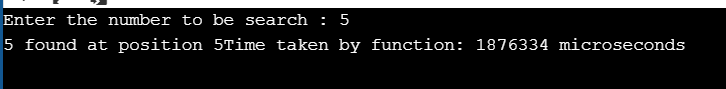
}

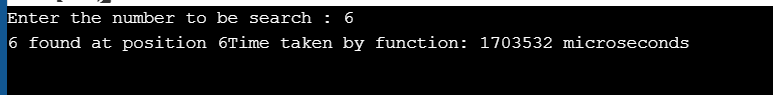
Hasil:

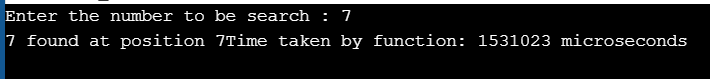


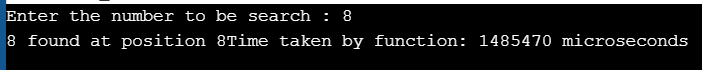


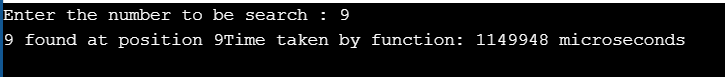


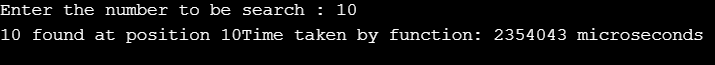












Pangkat Rekrusif:

#include <iostream>

#include <chrono>

using namespace std;

using namespace std::chrono;

int Pangkat(int y){

if (y==0){

return 1;

}

else{

return (2\*Pangkat(y-1));

}

}

int main() {

int p;

cout<<"masukkan pangkat : ";

cin>>p;

auto start = high\_resolution\_clock::now();

cout<<"Hasil = "<<Pangkat(p);

auto stop = high\_resolution\_clock::now();

auto duration = duration\_cast<nanoseconds>(stop-start);

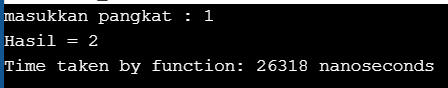
cout<<"\nTime taken by function: "

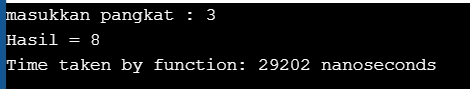
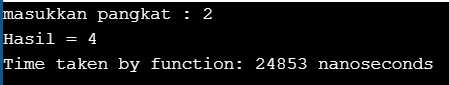
<< duration.count()<<" nanoseconds"<<endl;

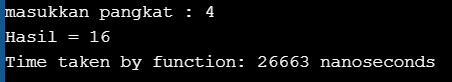
return 0;

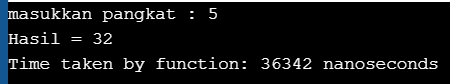
}

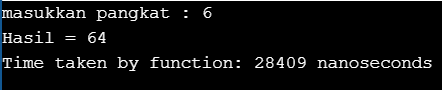
Hasil:

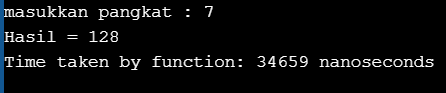


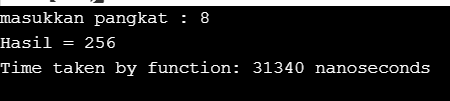


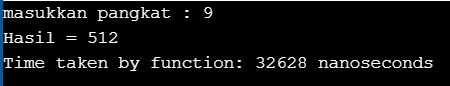


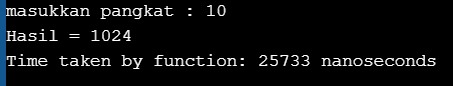












Pangkat Iterasi:

#include<iostream>

#include<conio.h>

#include <chrono>

using namespace std;

using namespace std::chrono;

int pangkat (int n)

{int hasil=1;

for (int i=1;i<=n;i++)

{

hasil=hasil\*2;

}

return hasil;

}

int main()

{

int j;

cout<<"Masukkan Pangkat : ";

cin>>j;

cout<<"Hasilnya = ";

auto start = high\_resolution\_clock::now();

cout<<pangkat(j);

auto stop = high\_resolution\_clock::now();

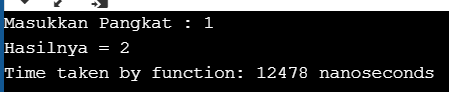
auto duration = duration\_cast<nanoseconds>(stop-start);

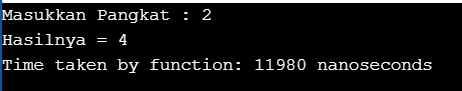
cout<<"\nTime taken by function: "

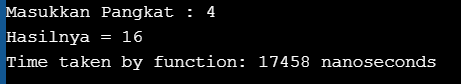
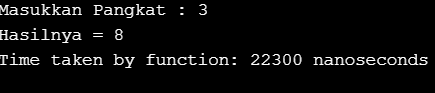
<< duration.count()<<" nanoseconds"<<endl;

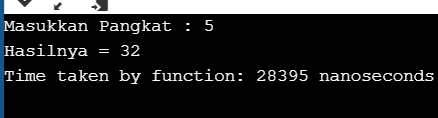
getch();

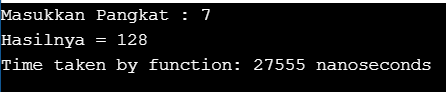
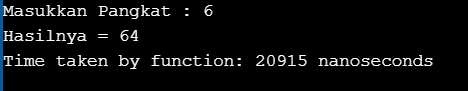
}Hasil :

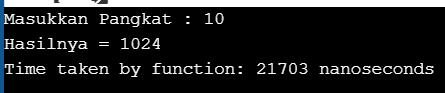
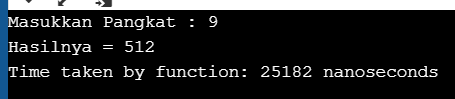
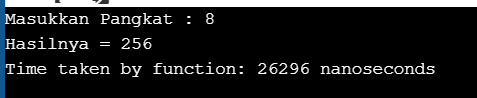












Analisa :

Pada grafik ini, waktu memiliki satuan microsecond. Dari grafik tersebut terlihat bahwa binary memiliki runtime terlama pada percobaan 4 sampai 6 dan percobaan 9. Sedangkan linear lebih stabil. Namun dari grafik diatas dan dengan runtime yang telah didapati merupakan tolak ukur yang kurang baik dalam menentukan keefektifan algoritma.

Pada grafik ini, waktu/runtime memiliki satuan nanosecond. Terlihat bahwa penggunaan rekrusif lebih cepat untuk perhitungan pangkat. Namun penyebaran garis yang tidak stabil memberi kesimpulan bahwa runtime tidak cocok untuk mengukur kefektifan suatu algoritma pangkat.