TUGAS ANALISIS ALGORITMA

Perbandingan Running Time



Oleh:

Intan Pratiwi 140810160022

Bening Kusumahati 140810160044

Hilya Tsaniya 140810160046

Program Studi S-1 Teknik Informatika

Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Padjadjaran

2019

* Program Pencarian nilai 2n rekursif

#include <iostream>

#include <cstdlib>

#include <chrono>

using namespace std;

using namespace std::chrono;

int pangkat(int a, int N ){

if (N==0){

return (1);

}

else{

return (a\*pangkat (a, N-1));

}

}

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

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

int b =2, x =3, hasil\_pangkat;

cout<<"Bilangan yang dipangkatkan : "<<b;

hasil\_pangkat = pangkat(b, x);

cout<<"x = "<<x<<endl<<endl;

cout<<"b^x = "<<b<<"^"<<x<<endl;

cout<<" = "<<hasil\_pangkat<<"\n";

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

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

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

}

* Program Pencarian nilai 2n iterative

#include<iostream>

#include <chrono>

using namespace std;

using namespace std::chrono;

int pangkat (int a, int b)

{int hasil=1;

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

{

hasil=hasil\*a;

}

return hasil;

}

int main()

{

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

int i =2,j=33;

cout<<"PROGRAM MENGHITUNG PANGKAT"<<endl;

cout<<endl<<"Angka : " <<i;

cout<<endl<<"Pangkat : "<<j;

cout<<endl<<"Hasilnya = ";

cout<<pangkat(i,j)<<"\n";

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

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

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

}

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

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

int b =2, x =3, hasil\_pangkat;

cout<<"Bilangan yang dipangkatkan : "<<b;

hasil\_pangkat = pangkat(b, x);

cout<<"x = "<<x<<endl<<endl;

cout<<"b^x = "<<b<<"^"<<x<<endl;

cout<<" = "<<hasil\_pangkat<<"\n";

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

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

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

}

* Analisis perbandingan Running Time kedua cara

|  |  |  |  |
| --- | --- | --- | --- |
| No | n | Iteratif (s) | Rekursif (s) |
| 1 | 3 | 0,000057 | 0,0005 |
| 2 | 9 | 0,000505 | 0,000506 |
| 3 | 13 | 0,000486 | 0,000501 |
| 4 | 19 | 0,000063 | 0,000057 |
| 5 | 25 | 0,000052 | 0,000051 |
| 6 | 27 | 0,000078 | 0,000336 |
| 7 | 29 | 0,000052 | 0,000051 |
| 8 | 31 | 0,000048 | 0,000056 |
| 9\* | 32 | 0,000051 | 0,000473 |
| 10\* | 33 | 0,000047 | 0,00006 |

Catatan : warna kuning waktu yang lebih besar

* Program Linear Search

arr = [ 2, 3, 4, 10, 40 ];

x = 10;

n = len(arr);

result = search(arr, n, x)

if(result == -1):

print("Element tidak ditemukan dalam array")

else:

print("Element ditemukan pada array di index ke", result);

end = time.clock()

print("\nRuntime = ",end-start, "seconds")

import time

start = time.clock()

for a in range(1000000):

b = 100\*a - 99

def search(arr, n, x):

for i in range (0, n):

if (arr[i] == x):

return i;

return -1;

arr = [ 2, 3, 4, 10, 40 ];

x = 10;

n = len(arr);

result = search(arr, n, x)

if(result == -1):

print("Element tidak ditemukan dalam array")

else:

print("Element ditemukan pada array di index ke", result);

end = time.clock()

print("\nRuntime = ",end-start, "seconds")

* Program Binary Search

arr = [ 2, 3, 4, 10, 40 ]

x = 10

result = binarySearch(arr, 0, len(arr)-1, x)

if result != -1:

print("Element ditemukan pada array di index ke % d" % result)

else:

print ("Element tidak ditemukan dalam array")

end = time.clock()

print("\nRuntime = ",end-start, "seconds")

import time

start = time.clock()

for a in range(1000000):

b = 100\*a - 99

def binarySearch (arr, l, r, x):

if r >= l:

mid = l + (r - l)//2

if arr[mid] == x:

return mid

elif arr[mid] > x:

return binarySearch(arr, l, mid-1, x)

else:

return binarySearch(arr, mid + 1, r, x)

else:

return -1

* Analisis Perbandingan Running time Linear Search dan Binary Search

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Percobaan | Banyak data | Linear (s) | Binary (s) |
| 1 | Data ke 1 | 10 | 0.175338 | 0.160676 |
| 2 | Data ke 5 | 10 | 0.158877 | 0.17045100000000002 |
| 3 | Data ke 10 | 10 | 0.157463 | 0.169197 |
| 4 | Data ke 1 | 50 | 0.176945 | 0.17588900000000002 |
| 5 | Data ke 12 | 50 | 0.192744 | 0.165658 |
| 6 | Data ke 25 | 50 | 0.169161 | 0.164324 |
| 7 | Data ke 37 | 50 | 0.170516 | 0.17472800000000002 |
| 8 | Data ke 50 | 50 | 0.172327 | 0.159989 |
| 9 | Data ke 1 | 70 | 0.173455 | 0.155016 |
| 10 | Data ke 17 | 70 | 0.161554 | 0.173248 |
| 11 | Data ke 35 | 70 | 0.14923799999999998 | 0.17716500000000002 |
| 12 | Data ke 52 | 70 | 0.16681200000000002 | 0.152502 |
| 13 | Data ke 70 | 70 | 0.16679 | 0.164746 |

* Kesimpulan

Pada algoritma pencarian nilai 2n antara rekursif dan iterative tidak ada perbedaan nilai running time yang signifikan, nilai running time kedua cara bisa dibilang hampir sama

Pada algoritma pencarian linear dan binary, untuk pencarian data yang berposisi di awal metode linear search memiliki running time lebih kecil dibanding binary search, namun untuk data berposisi di tengah terutama pada jumlah data yang banyak, running time binary search lebih kecil, meski memiliki algoritma yang lebih kompleks binary search lebih efektif dalam pencarian data besar dan banyak