**Code:**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Lab 08

\* Created by Laith Assaf

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <cmath>

using namespace std;

int linear(int n);

int logarithmic(int n);

int quadratic(int n);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* main()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main()

{ int a,n;

// Get N

cout << "Enter N: ";

cin >> n;

// Steps 1-4

/\*

a = linear(n);

cout << "Linear count = " << a << endl;

\*/

/\*

// Steps 5-8

a = logarithmic(n);

cout << "Logarithmic count = " << a << endl;

\*/

// Steps 9-12

a = quadratic(n);

cout << "Quadratic count = " << a << endl;

// Success

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* linear()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int linear(int n)

{ int i;

float p;

int counter;

counter = 0;

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

{ cout << i << " ";

p = pow(2.718F,i);

cout << p << endl;

counter+= 3;

};

return counter;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* logarithmic()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int logarithmic(int n)

{ int sum;

int counter;

counter = 0;

sum = 1;

while(sum<n)

{ sum = sum\*2;

cout << sum << endl;

counter+= 2;

};

return counter;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* quadratic()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int quadratic(int n)

{ int i,j,m;

int inner,outer;

// Initialize

inner = outer = 0;

// Get m

cout << "Enter M: ";

cin >> m;

// Outer loop

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

{

// Inner loop

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

{ cout << '#';

inner++;

};

cout << endl;

outer++;

};

//Display Outer

cout << "Outer count = " << outer << endl;

return inner;

}

**Questions:**

3.

|  |  |  |  |
| --- | --- | --- | --- |
| N | 10 | 100 | 1000 |
| Count | 30 | 300 | 3000 |

**The ratio is 3 for all three which like the O(n) which is O(1) which is constant.**

7.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | 10 | 100 | 1000 | 10,000 |
| Log2N | 3.32 | 6.64 | 9.97 | 13.29 |
| Count | 8 | 14 | 20 | 28 |

**2.41, 2.11, 2.01, 2.11 are the ratios which are pretty constant, which confirms that the ratio matches O(LogN) complexity.**

11.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | 10 | 20 | 100 | 200 |
| Count | 110 | 420 | 10100 | 40200 |

**The counts are close to 1. They become even closer to 1 as N increases. The count grows proportionally with N^2.**