Análisis de Algoritmos

Actividad de seguimiento – Parte 1

Esta actividad tiene como propósito que el estudiante realice el análisis de tiempo de ejecución para algoritmos iterativos.

Nombre de los integrantes del grupo:

* Estudiante 1: Alisson Campos Marin
* Estudiante 2: Johan Andrey Ortiz

Determine el tiempo de ejecución para los siguientes algoritmos iterativos.

public void metodo1 ( ){

int temp = 0; **→ 1**

int x = 0; **→ 1**

int i = 64; **→ 1**

while( i > 0 ) { **→ long2(i)+1**

x++; **→ 1**

temp++; **→ 1**

i = i / 2; **→ 1**

}

}

**T (n) = 4(log2(i) + 1) + 1 + 3**

**T (n) = 4(log2(i) + 1) + 4**

public void metodo2( int n ){ **→ 1**

int y = 0; **→ 1**

int w = 2; **→ 1**

for ( int i = 0 ; i < n ; i++ ){ **→ 2n**

for ( int j = 0 ; j < n; j++ ){ **→ 2n**

w++; **→ n**

for ( int k = n ; k > 0 ; k-- ){ **→ 2n**

y++; **→ n**

}

}

}

}

**T (n1) = 3n + 1 + 1 = 3n + 2**

**T (n2) = 3n + 1 + 1 = 3n + 2**

**T (n3) = 2n + 1 + 1 = 2n + 2**

**T (n) = {[(3n + 2)** **\* n] + 3n + 2} \* n + 2n + 2 + 3**

**T (n) = {3n^2 + 2n** **+ 3n + 2} \* n + 2n + 5**

**T (n) = 3n^3 + 5n^2** **+ 4n + 5**

public void metodo3 ( ){

int x = 0, y = 1, w = 0, i; **→ 4**

i = 128;

while( i > 0 ){ **→ 4(log2(i) + 1) + 1 = 4log2(i) + 5**

x++;

y = y \* 1;

i = i / 2;

}

for ( i = n ; i > 0 ; i-- ){ **→ 4n + 1 + 1 = 2n + 2**

w++;

y++;

}

}

**T (n) = 4log2(i) + 5 + 4n + 2**

**T (n) = 4log2(i) + 4n + 7**

public void metodo4 ( int n ){ **→ 1**

int x, i, j, y; **→ 4**

x = 0;

y = 1;

for ( i = 0 ; i < n ; i++ ){ **→ 3n**

j = 1;

while (j < n){ **→ 3log2(n)**

x = x+2;

j\*=2

}

y++;

}

}

**T (n) = 3log2(n) + 1 + 3n + 5**

**T (n) = 3log2(n) + 3n + 6**

public void metodo5( int n ){ **→ 1**

int x, i, j; **→ 3**

n = 1;

x = 0;

for ( i = 0 ; i < n ; i++ ){ **→ 3n+1**

for ( j = 1 ; j < n ; j\*=2 ){ **→ 3log2(n)+1**

x = x+2;

}

m++;

}

}

**T (n) = (3log2(n) + 1) \* n + 3n + 4**

public void metodo6 ( int n ){ **→ 1**

int t = 0, i, j, l; **→ 4**

j = 256;

while( j > 1 ){ **→ 3(log2(j) + 1) + 1 = 3log(j) + 4**

t++;

j = j / 2;

}

for (i = 2; i <= n+3; i++){ **→ 2n + 1 \* (n + 3)**

for (j = n; j >= 0; j--){ **→ (3n - 1 + 1) \* n**

t--;

}

}

for (l = n; l > 0; l--) { **→ 3n+1**

t--;

}

}

**T (n) = 1 + 4 + (3log(j) + 4) + [(3n - 1 + 1) \* n + 2n + 1 \* (n + 3)]**

**T (n) = 9 + 3log(j) + [3n^2 + 2n + n + 3]**

**T (n) = 9 + 3log(j) + 3n^2 + 3n + 3~~~~~~**

Para el cálculo del tiempo de ejecución del siguiente algoritmo, tenga en cuenta que el condicional if(m % n == 0) sólo es verdadero en la última iteración.

for (int j = 2; j <= (n - 1); j++){

m = div;

if (m % n == 0){

switch (1){

case 0: *metodo1*(); → **T(n)= n + 3**

case 1: *metodo2*(); → **T(n)= 2n + 5**

for (int k = 0; k < n; k++){ → **(3n+2)\*(n-1)**

m++; **= 3n^2-3n+2n-2 = 3n^2-n-2**

}

case 2: *metodo3*(); → **T(n)= n + 4**

}

break;  **→ 1**

}

div ++;

}

Asuma que los tiempos de ejecución para los métodos son:

Metodo1() ---> T(n)= n + 3 Metodo2() ---> T(n)= 2n + 5 Metodo3() --->T(n)= n + 4

**T (n) = 3n^2 + 6n + 8**

Estime el tiempo de ejecución T(n), del siguiente fragmento de código. Asuma que las variables se encuentran declaradas e inicializadas. Para el cálculo del tiempo de ejecución, asuma que el condicional if (i % n == 2) es verdadero solamente en la última iteración del ciclo.

for (int j = 1; j <= (n - 3); j++)

{

s++;

i = i +1;

if (i % n == 2)

{

switch (1)

{

case 0: metodo1();

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

{

s++;

}

case 1: metodo2();

metodo3();

for (int k = n; k > 0; k/=2)

{

s++;

}

case 2: metodo4();

case 3: metodo5();

break;

case 4: metodo6();

}

break;

}

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

{

s++;

}

s--;

}

Asuma que los tiempos de ejecución para los métodos son:

* metodo1() ---> T(n)= n + 3
* metodo2() ---> T(n)= n2 + 5
* metodo3() ---> T(n)= n3 + n
* metodo4() ---> T(n)= n \* log(n)
* metodo5() ---> T(n)= 2n + 4
* metodo6() ---> T(n)= 3n \* log(n)

**Switch T (n) = [3(n - k) + 2]\*(n-1) + n^2 + 5 + n^3 + n + [3log2(n) + 5]\*(n-1) + n \* log(n) + 2n + 4 + 1**

**If T(n) = n + 1 + Switch T (n)**

**T(n) = If T(n) + Switch T (n)**

public void metodo9 ( int n, int b, int c)

{

if (n>b)

{

if (n>c)

{

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

{

metodo1();

}

}

else

{

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

{

metodo2();

}

}

}

else

{

metodo3();

}

}

metodo1() ---> T(n)= 3n + 100 metodo2() ---> T(n)= 3n + 120

metodo3() ---> T(n)= 2n2 + 10

**T (n) = 7n + 127**

public void metodo10 ( int n ){

int x, i, j;

m = 1;

x = 0;

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

for ( j = 1 ; j < n ; j\*=2 ){

x = x+2;

}

m++;

x--;

}

for ( i = 2 ; i <= n ; i++ ){

for ( j = 1 ; j < n ; j\*=2 ){

x = x+2;

}

}

}

**T (n) = (3log2(n) + 1)(n - 1) + (3nlog2(n) + 1)\*n + 6n + 5**