Trabajo. Análisis de complejidad

**Objetivo**

El objetivo de esta actividad es evaluar la complejidad de algoritmos con las técnicas estudiadas.

**Descripción**

Dadas las siguientes funciones, realiza un análisis de su complejidad indicando la operación básica, su orden de crecimiento y el coste de ejecución. En caso de ser necesario, distingue entre caso peor, caso medio y caso mejor.

.

**Algoritmo 1:**

sum = 0;

for(i=0;i<sqrt(n)/2;i++)

sum++;

for(j=0 ;j<sqrt(n)/4;j++)

sum++;

for(k=0;k<8+j;k++)

sum++;

**Algoritmo 2:**

sum = 0;

for(i=0;i<sqrt(n)/2;i++)

for(j=i;j<8+i;j++)

for(k=j;k<8+j;k++)

sum++;

**Algoritmo 3:**

sum = 0;

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

for(j=1;j<i\*i;j++)

for(k=1;k<j;k++)

if (j % i == 1)

sum++;

**Algoritmo 4:**

int f(int n) {

if (n<=1)

return 1;

else

return n\*f(n/2);

}

**Solución:**

Dadas las siguientes funciones, realiza un análisis de su complejidad indicando:

* La operación básica
* Su orden de crecimiento
* El coste de ejecución.

En caso de ser necesario, distingue entre caso peor, caso medio y caso mejor.

1. **Algoritmo 1**

sum = 0;

for(i=0;i<sqrt(n)/2;i++)

sum++;

for(j=0 ;j<sqrt(n)/4;j++)

sum++;

for(k=0;k<8+j;k++)

sum++;

* ***La operación básica:***

;

* ***El coste de ejecución:***

|  |  |  |
| --- | --- | --- |
| 1 |  | 1 asignación |
| 2 |  |  |
| 3 |  | 1 incremento |
| 4 |  |  |
| 5 |  | 1 incremento |
| 6 |  |  |
| 7 |  | 1 incremento |

Resolviendo cada una de las sumatorios descritas en la tabla superior:

Como el valor de depende de la sumatoria anterior se puede concluir que:

Sustituyendo la sumatoria:

* ***Su orden de crecimiento:***

1. **Algoritmo 2**

sum = 0;

for(i=0;i<sqrt(n)/2;i++)

for(j=i;j<8+i;j++)

for(k=j;k<8+j;k++)

sum++;

* ***La operación básica:***

;

* ***El coste de ejecución:***

|  |  |  |
| --- | --- | --- |
| 1 |  | 1 asignación |
| 2 |  |  |

Resolviendo:

* ***Su orden de crecimiento:***

1. **Algoritmo 3**

sum = 0;

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

for(j=1;j<i\*i;j++)

for(k=1;k<j;k++)

if (j % i == 1)

sum++;

* ***La operación básica:***
* ***El coste de ejecución:***

|  |  |  |
| --- | --- | --- |
| 1 |  | 1 asignación |
| 3 |  |  |

Resolviendo:

* ***Su orden de crecimiento:***

1. **Algoritmo 4**

int f(int n) {

if (n<=1)

return 1;

else

return n\*f(n/2);

}

* ***La operación básica:***
* ***El coste de ejecución:***
* ***Su orden de crecimiento:***

**Entrega de la actividad**

Entrega un fichero con las respuestas de cada apartado. Es importante justificar detalladamente la selección de las fórmulas.

Solo se aceptan ficheros editables (.doc, .docx, .odf, .rtf). No se aceptan imágenes escaneadas del ejercicio ni en formato .pdf.

**Revisión de la actividad**

La revisión de las actividades calificables no se realizará por el foro, sino a través del tutor.

**Rúbrica**

|  |  |  |  |
| --- | --- | --- | --- |
| Trabajo 1 | Descripción | Puntuación máxima  (5 puntos) | Peso  % |
| Criterio 1 | La representación matemática es correcta. | 2,5 | 50% |
| Criterio 2 | Las fórmulas están bien desarrolladas  y el orden de complejidad es correcto. | 1,5 | 30% |
| Criterio 3 | El coste computacional es correcto. | 1 | 20% |

La existencia de fallos excesivos o las apreciaciones especialmente interesantes pueden modificar esta rúbrica con puntos negativos o superiores a la máxima.

**Extensión máxima:** 2 páginas.