Analsis y diseño de algoritmos

Unidad I
Analisis de Complejidad Computacional
Practica 01: Analisis de Casos

Docente:
M. en C. Erika Sanchez-Fermat

Unidad Profesional Interdedisciplinaria de Ingenieria Campus Zacatecas
Instituto Politecnico Nacional
Septiembre 2023

1. Objetivo de la práctica

El objetivo principal de esta práctica es que los estudiantes adquieran una comprensión sólida y profunda de los conceptos clave relacionados con la complejidad computacional de los algoritmos. Los tres casos principales a analizar, mejor caso, peor caso y caso promedio, permiten una evaluación completa del rendimiento de un algoritmo bajo diferentes circunstancias.

1.1. Objetivos particulares

- **Comprender la Variabilidad de la Ejecución:** Los algoritmos pueden comportarse de manera diferente según las características de los datos de entrada. Al analizar el mejor, peor y caso promedio, los estudiantes aprenderán a apreciar cómo un algoritmo puede variar en su rendimiento.
- **Aplicar Conceptos de Complejidad Computacional:** La práctica ayudará a los estudiantes a aplicar conceptos fundamentales de la teoría de la complejidad computacional, como la notación Ogrande (Big O), para describir y comparar el rendimiento de los algoritmos..
- **Desarrollar Habilidades de Análisis Crítico:** A través del análisis de diferentes casos, los estudiantes mejorarán sus habilidades de análisis crítico, aprendiendo a identificar situaciones en las que un algoritmo puede ser más eficiente o ineficiente.
- **Preparación para Desarrollo de Algoritmos Eficientes:** Al comprender los casos de mejor, peor y caso promedio, los estudiantes estarán mejor preparados para diseñar y seleccionar algoritmos eficientes en situaciones reales, lo que es fundamental en la resolución de problemas computacionales.
- **Promover la Resolución de Problemas:** A través de la práctica, los estudiantes aprenderán a abordar y resolver problemas computacionales de manera más efectiva, seleccionando o adaptando algoritmos en función de las restricciones de tiempo y recursos.

2. Desarrollo de la práctica

2.1. implementación de los algoritmos

Escribe en python los siguientes métodos de ordenamiento:

- Burbuja
  El metodo de burbuja consiste en recorrer, un numero de veces los elementos de la lista, realizando la comparacion entre cada par de elementos pertenecientes al arreglo. De tal manera que si el elemento de la izquierda es mayor que el de la derecha, se realiza un intercambio de posicion.

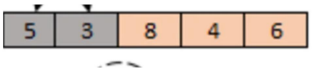

Figura 1: Se hace una comparacion entre el primer y segundo elemento de la lista, si el elemento de la izquierda es mayor que el de la derecha, se realiza un intercambio de posición.

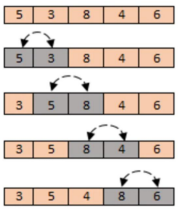

Figura 2: Se repite el mismo procedimiento con cada pareja del arreglo hasta que el arreglo queda totalmente en orden.

- Burbuja Optimizada

El metodo de burbuja optimizado implementa una bandera, que reduce significativamente el numero de comparaciones. Es una mejora, debido a que el algoritmo termina cuando todos los elementos están ordenados.

Implementación en Python:

```
def burbuja_optimizada(array):
    for i in range(len(array)):
        intercambio= False
        for j in range (len(array)-i-1):
            if array[j]>array[j+1]:
                aux=array[j]
                array[j]=array[j+1]
                array[j+1]=aux
                intercambio = True
        if intercambio == False:
            break
```

Figura 3: Implementacion en phyton con una bandera.

2.2. Analisis de Casos

Desarrollar un reporte con lo siguiente:

- Calcular el mejor, peor y caso promedio para ambos algoritmos de ordenamiento, dando 3 ejemplos de datos de entrada para cada caso.

  Mejor caso Burbuja:

  1. A=1,2,3,4,5
  2. B=1,10,30,32,45
  3. C=1,2

  Peor caso Burbuja:

  1. A=5,4,3,2,1
  2. B=10,6,3,2,1
  3. C=45, 32, 29, 26, 17, 15, 12

  Caso promedio Burbuja:

  1. A=3,1,4,10,2
  2. B=10,5,2,29,1
  3. C=45, 67, 2, 25, 17, 43, 5

  Mejor caso Burbuja mejorada:

  1. A=10,20,30,40,50
  2. B=10,10,25,32,48
  3. C=5,8

  Peor caso Burbuja mejorada:

  1. A=50,49,32,23,12
  2. B=100,65,32,21,14
  3. C=48, 35, 22, 21, 17, 16, 12

  Caso promedio Burbuja mejorada:

  1. A=3,1,4,10,2

2. B=10,5,2,29,1
3. C=45, 67, 2, 25, 17, 43, 5

- Encontrar y justificar a cuál clase de las siguientes pertence cada caso de cada algoritmo:

Brubuja:

- O(n)2: En este caso, el metodo de ordenamiento de burbuja pertenece a está complejidad O cuadrática cuanddebido a que debe realizar una iteración para cada elemento del arreglo se encuentra desordenado o odenado de manera inversa. Por lo que el peor caso tiene está complejidad.

- O(n)2: En este caso, el metodo de ordenamiento de burbuja pertenece a complejidad O cuadrática cuando el arreglo se encuentra desordenado. Por lo que tiene que realizar un elevado número de iteraciones,el caso promedio tiene está complejidad., por lo que en su peor y caso promedio entra en esta categoria.

- O(n): En este caso, entra el mejor de los casos para le método de burbuja, debido a que si de entrada, ya contamos con el arreglo ordenado, el algoritmo no tendra que realizar mas iteraciones para ordenar el arreglo.

Brubuja mejorada:

- O(n): En el caso del método de burbuja mejorado, el mejor caso se encuentra cuando la lista ya se encuentra ordenada, por lo que el algoritmo no necesitará realizar un mayor número de iteraciones.

- O(n)2: En el caso del método de burbuja mejorado, el peor caso se encuentra cuando la lista no se encuentra ordenada o se encuentra ordenada de manera inversa por lo que el algoritmo necesitará realizar un mayor número de iteraciones.

- O(n)2: En el caso del método de burbuja mejorado, el caso promedio se encuentra cuando la lista no se encuentra ordenada por lo que el algoritmo necesitará realizar un elevado número de iteraciones.

2.3. Comparación de Resultados

Comparar el tiempo de ejecución de cada algoritmo de ordenamiento para el mejor, peor y caso promedio. ¿Qué conclusiones puedes sacar de estos resultados?

- **Brubuja:** El tiempo promedio de ejecucion fue de 7.67 milisegundos

- **Brubuja mejorada:** El tiempo promedio de ejecucion fue de 3.67 milisegundos
- **Conclsusión:**

  El método de la burbuja mejorado es mucho mas eficiente que el método común, sin embargo no es tan notoria la eficiencia, por lo que se puede decir que la complejidad de ambos metodos no varia y no cambia mucho una con respecto de la otra.

3. Conclusiones

- El método de busqueda de la burbuja, es un método muy ineficiente,que tiende a incrementar mucho su tiempo de ejecución, por lo que otros métodos de búsqueda son mas efectivos que el de la burbuja.
- El método de la burbuja mejorada y el método normal,son muy parecidos entre ambos y ambos tienen la misma complejidad en sus mejores y peores casos, por lo que se puede decir que no hay mucha diferencia entre ambos métodos.
- A pesar de no ser el método mas eficiente, es un método muy facil de implementar, además de ser muy sencillo de comprender, por lo que para listas sencillas no muy desordenadas o para aprender algoritmos iterativos es un método viable

4. Referencias

- Método de burbuja mejorado. (s/f). Prezi.com. Recuperado el 19 de septiembre de 2023, de https://prezi.com/p/g6sxlqrn-zrt/metodo-de-burbuja-mejorado/
- Amorin, D. (2022, abril 21). Ordenamiento de Burbuja en Python. Diego Amorin. https://diegoamorin.com/ordenamiento-burbuja-python/
- Computacion I. (s/f). Uchile.cl. Recuperado el 19 de septiembre de 2023, de https://users.dcc.uchile.cl/teu/CC10A/Apuntes/ordenamiento/index.html
- Alarcón, J. M., JMAlarcon. (s/f). Rendimiento de algoritmos y notación Big-O. campusMVP.es. Recuperado el 19 de septiembre de 2023, de https://www.campusmvp.es/recursos/post/Rde-algoritmos-y-notacion-Big-O.aspx