

Entrega un reporte de solución por escrito a cada uno de los siguientes problemas.

1. El algoritmo del **Bubble Sort** es muy popular por su simplicidad. Si lo recuerdas, la idea de este algoritmo consiste en recorrer todo el arreglo de datos, comparando cada elemento con su sucesor, y en caso de que estos 2 datos estén desordenados, intercambiarlos. El final del recorrido de todo el arreglo, se asegura que el elemento más grande (si se está ordenando ascendentemente) ha quedado en la última posición. Por lo tanto, el algoritmo tiene que recorrer de nuevo el arreglo desde el inicio con el mismo proceso para dejar el siguiente elemento mayor en la penúltima posición. Obviamente, este proceso se repite hasta que se ordenan los últimos 2 datos en el arreglo.

- (10 puntos) En base a esta descripción del Bubble Sort, escribe en pseudocódigo (o en el lenguaje de tu preferencia) el algoritmo computacional que describe a este proceso.
- (5 puntos) Explica si este algoritmo se puede analizar obteniendo el comportamiento para el peor caso, el caso promedio y el mejor caso, o bien, si es un algoritmo que tiene un comportamiento igual para todos los casos ("every case"). Justifica.
- (10 puntos) Con las herramientas de análisis general que hemos visto en clase, obtén el orden de este algoritmo. Trata de justificar lo mejor posible tu respuesta.
- (10 puntos) Aplica el algoritmo sobre un arreglo que ya está ordenado. ¿Qué observas? Como te darás cuenta, el algoritmo trabaja sin sentido para ordenar algo que ya está ordenado. Puesto que esto es crítico, un Bubble Sort mejorado podría considerar que cuando no hay intercambios en el barrido del arreglo, éste ya está ordenado, y por lo tanto se puede terminar el proceso de ordenamiento. Modifica el algoritmo que escribiste anteriormente, para que en una nueva versión se considere esta mejora.
- (5 puntos) De nuevo responde los incisos c) y d) pero ahora para la nueva versión del algoritmo que obtuviste en el inciso e).

2.

- (10 puntos) Escribe un algoritmo eficiente para buscar un dato en una matriz M de n X n. La matriz contiene a los datos ordenados por renglones de tal manera que:

$$M[i,j] < M[i,j+1] , M[i,j] < M[i+1, j] \text{ y } M[i,n] < M[i+1, 1]$$

El algoritmo deberá tener al menos un comportamiento de orden lineal y no cuadrático.

- (5 puntos) Para una cantidad de n2 datos ordenados, explica si este algoritmo pudiera resultar mejor que la búsqueda binaria en el peor caso. Justifica tu respuesta.

3. Dada la siguiente lista de datos: **87 23 12 15 76 91 44 56**

- (7 puntos) Muestra, paso a paso, la forma en que se hacen las modificaciones a la lista si se aplica el algoritmo del Merge Sort para ordenarla descendientemente (de mayor a menor).
- (8 puntos) Muestra, paso a paso, la forma en que se hacen las modificaciones a la lista si se aplica el algoritmo del Quick Sort para ordenarla descendientemente (de mayor a menor) considerando como elemento pivote al último elemento en la lista.
- (10 puntos) Para este caso particular, compara el comportamiento de ambos algoritmos, contando la cantidad de veces que se ejecuta la operación básica, que en este caso es la comparación de datos en los subalgoritmos Une y Partición presentados en clase. ¿Cuál algoritmo resultó ser mejor para este caso?
- (5 puntos) Menciona un ejemplo de una secuencia de 8 datos en que el algoritmo del Merge Sort se comporte mejor que el Quick Sort para ordenarlos ascendentemente.

4. (15 puntos) Utilizando la técnica de divide y vencerás plantea un algoritmo que sirva para encontrar el elemento menor de un arreglo de datos desordenados. Escribe el algoritmo computacional en pseudocódigo. Haz un análisis general y explica si el planteamiento de solución con la técnica de divide y vencerás representa alguna ventaja con respecto al algoritmo que en forma tradicional se emplearía para realizar este proceso. Justifica tu respuesta.