

Hoja de Trabajo HT-M2-1 – Solución

Sección 4 - EDA 2025-20

Preguntas de ordenamiento

1. Se tiene el siguiente arreglo de enteros: $[1, 2, 3, 4, 5, 6, 7, 8, 9]$. Se utiliza un algoritmo de ordenamiento desconocido, cuya complejidad temporal para hacer el ordenamiento fue cuadrática. ¿Cuál fue el algoritmo de ordenamiento usado?
- A) Merge Sort
 - B) Insertion Sort
 - C) Selection Sort
 - D) Ninguno de los anteriores

Explicación:

Opción C: El arreglo está originalmente ordenado, es decir, el mejor caso. Entre las opciones presentadas, Selection Sort es el único que puede tener una complejidad cuadrática en el mejor caso.

2. Los siguientes son puntos (X, Y) en un plano cartesiano: $[(1, 2), (1, 0), (5, 3), (0, 2)]$. Los puntos se ordenan de forma ascendente por la coordenada X con un algoritmo de ordenamiento ESTABLE. El resultado es:
- A) $[(1,0), (0,2), (1,2), (5,3)]$
 - B) $[(1,0), (1,2), (0,2), (5,3)]$
 - C) $[(0,2), (1,0), (1,2), (5,3)]$
 - D) $[(0,2), (1,2), (1,0), (5,3)]$

Explicación:

Opción D: Los puntos están ordenados y se respetó el orden relativo.

Las otras opciones son incorrectas porque:

$[(1,0), (0,2), (1,2), (5,3)]$

$[(1,0), (1,2), (0,2), (5,3)]$

$[(0,2), (1,0), (1,2), (5,3)]$

3. Dado el arreglo $[18, 12, 20, 30, 11, 32, 34, 29]$ se aplica el algoritmo Selection Sort para ordenarlo de mayor a menor. ¿En cuál iteración el arreglo estará completamente ordenado?
- A) 4
 - B) 5
 - C) 6
 - D) 7

Explicación:

Opción C:

Iteración 1: $[34, 12, 20, 30, 11, 32, 18, 29]$

Iteración 2: $[34, 32, 20, 30, 11, 12, 18, 29]$

Iteración 3: $[34, 32, 30, 20, 11, 12, 18, 29]$

Iteración 4: [34, 32, 30, 29, 11, 12, 18, 20]

Iteración 5: [34, 32, 30, 29, 20, 12, 18, 11]

Iteración 6: [34, 32, 30, 29, 20, 18, 12, 11]

Iteración 7: [34, 32, 30, 29, 20, 18, 12, 11]

Iteración 8: [34, 32, 30, 29, 20, 18, 12, 11]

4. Supongamos que se está ordenando un arreglo de ocho enteros utilizando Quicksort, y se acaba de terminar la primera partición y el arreglo quedó de la siguiente manera: [2, 6, 1, 7, 9, 12, 16, 10]. ¿Qué afirmación es correcta?
- A) El pivote podría ser el 7, pero no es el 9
 - B) El pivote no es el 7, pero podría ser el 9
 - C) El pivote podría ser el 7 o el 9
 - D) Ninguna de las anteriores

Explicación:

Opción C: El pivote podría ser el 7 o el 9. Para ambos casos, el pivote estaría correctamente ubicado (menores a la izquierda, mayores a la derecha).

5. Se tiene el siguiente arreglo de enteros: [1, 5, 10, -1, 4]. Se usa Shell Sort con valores de h fijos en este conjunto: 4, 2, 1 (se inicia en h=4). ¿Cuál de las siguientes opciones representa los resultados intermedios para el valor de h dado?
- A) [-1, 5, 10, 1, 4], h: 4 -> [1, -1, 4, 5, 10], h: 2 -> [-1, 1, 4, 5, 10], h: 1
 - B) [1, 5, 10, -1, 4], h: 4 -> [1, 5, 4, -1, 10], h: 2 -> [-1, 1, 4, 5, 10], h: 1
 - C) [1, 5, 10, -1, 4], h: 4 -> [1, -1, 4, 5, 10], h: 2 -> [-1, 1, 4, 5, 10], h: 1
 - D) [1, 5, 10, -1, 4], h: 4 -> [1, 5, 4, -1, 10], h: 2 -> [1, 1, 4, 5, 10], h: 1

Explicación:

Arreglo inicial = [1, 5, 10, -1, 4]

h=4:

- Se compara 1 con 4, no hay cambios

h=2:

- Se compara 1 con 10, no hay cambios

- Se compara 5 con -1, se intercambian

Arreglo modificado = [1, -1, 10, 5, 4]

- Se compara 10 con 4, se intercambian

Arreglo modificado = [1, -1, 4, 5, 10]

h=1:

- Se compara 1 con -1, se intercambian

Arreglo modificado = [-1, 1, 4, 5, 10]

- Todas las comparaciones restantes no generan cambios

6. Se tiene un conjunto de 15.000 datos, ¿para qué valores de h se ejecuta Shell Sort (con función de Knuth)?
- A) 9841,3280,1090,364,121,40,13,4,1
 - B) 9842,3287,1093,364,121,40,13,4,1
 - C) 9341,2280,1093,364,121,40,13,4,1
 - D) 9841,3280,1093,364,121,40,13,4,1

Explicación:

Opción D:

$h = 1$ (Shell Sort termina cuando h es igual a 1)

$h = 3g + 1 = 4$ (g siempre es el h previo, aquí sería igual a 1)

$h = 3g + 1 = 13$

$h = 3g + 1 = 40$

$h = 3g + 1 = 121$

$h = 3g + 1 = 364$

$h = 3g + 1 = 1093$

$h = 3g + 1 = 3280$

$h = 3g + 1 = 9841$

$h = 3g + 1 = 29524$

29524 es mayor que 15000, por lo que no se incluye en la secuencia.

Preguntas de tablas de símbolos

7. ¿Cuál de las siguientes afirmaciones NO es cierta sobre una tabla de hash, utilizando como resolución de conflictos Linear Probing, con factor de carga igual o menor a 0.5, suponiendo distribución uniforme de las llaves?
- A. El tamaño de la tabla debe ser de preferencia un número primo.
 - B. La tabla puede tener duplas (llave, valor) repetidas en su valor.
 - C. Todas las posiciones de la tabla tienen duplas (llave, valor) almacenadas.
 - D. El tamaño de la tabla debe ser preferiblemente el número primo mayor al doble del número de duplas (llave, valor) a almacenar.

Explicación:

Opción C: Si todas las posiciones de la tabla tienen duplas, el factor de carga no podría ser del 50%.

8. Considere una tabla hash de tamaño 10, usando la función hash $H(k) = k \bmod 10$ y el sondeo lineal como método de resolución de colisiones. Al insertar las llaves 15, 25, 35, 45, 55, ¿cuál es la afirmación VERDADERA sobre la posición final de las llaves después de insertarlas todas?
- A. La posición final de la llave 55 es 5
 - B. La posición final de la llave 45 es 9
 - C. La posición final de la llave 25 es 6
 - D. La posición final de la llave 35 es 5

Explicación:

Opción C:

$\text{hashValue}(15) = 15 \bmod 10 = 5$

$\text{hashValue}(25) = 25 \bmod 10 = 5 \rightarrow$ Debido al sondeo lineal, la posición final es 6

$\text{hashValue}(35) = 35 \bmod 10 = 5 \rightarrow$ Debido al sondeo lineal, la posición final es 7

$\text{hashValue}(45) = 45 \bmod 10 = 5 \rightarrow$ Debido al sondeo lineal, la posición final es 8

$\text{hashValue}(55) = 55 \bmod 10 = 5 \rightarrow$ Debido al sondeo lineal, la posición final es 9

9. Considere una tabla de hash utilizando Linear Probing con capacidad $M = 10$. Se insertan los siguientes elementos usando la función de hash $h(key) = (|key| \% M)$, $elements = [(-2, a), (-1, b), (-5, a), (4, c), (-9, d), (3, e), (-7, f), (8, f), (9, h), (6, i)]$, ¿cuál es el estado final de la tabla si se ignora cualquier operación de rehash y factor de carga?
- A. 0: (9, d), 1: (-1, b), 2: (-2, a), 3: (3, e), 4: (4, c), 5: (-5, a), 6: (6, i), 7: (-7, f), 8: (8, f), 9: (9, h)
 B. 0: (-9, d), 1: (-1, b), 2: (-2, a), 3: (3, e), 4: (4, c), 5: (-5, a), 6: (6, i), 7: (-7, f), 8: (8, f), 9: (-9, h)
 C. 0: (9, h), 1: (-1, b), 2: (-2, a), 3: (3, e), 4: (4, c), 5: (-5, a), 6: (6, i), 7: (-7, f), 8: (8, f), 9: (-9, d)
 D. 0: (9, h), 1: (-1, b), 2: (-2, a), 3: (3, e), 4: (4, c), 5: (5, a), 6: (6, i), 7: (7, f), 8: (8, f), 9: (-9, d)

Explicación:

Opción C:

$h(-2) = |-2| \% 10 = 2$. Inserta (-2, a) en la posición 2

$h(-1) = |-1| \% 10 = 1$. Inserta (-1, b) en la posición 1

$h(-5) = |-5| \% 10 = 5$. Inserta (-5, a) en la posición 5

$h(4) = |4| \% 10 = 4$. Inserta (4, c) en la posición 4

$h(-9) = |-9| \% 10 = 9$. Inserta (-9, d) en la posición 9

$h(3) = |3| \% 10 = 3$. Inserta (3, e) en la posición 3

$h(-7) = |-7| \% 10 = 7$. Inserta (-7, f) en la posición 7

$h(8) = |8| \% 10 = 8$. Inserta (8, f) en la posición 8

$h(9) = |9| \% 10 = 9$. Colisión. El sondeo encuentra a la posición 0 como libre. Inserta (9, h) en la posición 0

$h(6) = |6| \% 10 = 6$. Inserta (6, i) en la posición 6

10. Para una tabla de hash vacía de capacidad $M = 13$, que utiliza el mecanismo Linear Probing para el manejo de colisiones, ¿Cuántas llaves diferentes se alcanzan a insertar antes de que se exceda el factor de carga de 0.75?
- A. 7
 B. 8
 C. 9
 D. 10

Explicación:

Opción C:

Capacidad de la tabla = 13

Límite de carga = $0.75 \times 13 = 9.75$

Antes de pasarse, solo se pueden insertar 9 llaves ($9/13 = 0.69 < 0.75$).

Con 10 llaves ya se excede ($10/13 = 0.77 > 0.75$).

11. ¿Cuál de las siguientes afirmaciones es **FALSA** para una tabla de hash que usa linear probing?
- A. El factor de carga de la tabla hash puede calcularse como $\alpha = N/M$, con N como el número de pares llave-valor y M el tamaño de la tabla.
 B. La operación de rehashing se hace si el factor de carga supera un umbral o factor de carga máximo.
 C. Las casillas marcadas en las operaciones de eliminación son contadas en el total de elementos, aumentando el factor de carga.
 D. El arreglo podría contener múltiples posiciones marcadas por previas operaciones de eliminación y estas no se considerarían en una operación de rehashing.

Explicación:

Opción C: Las casillas marcadas no cuentan como parejas llave-valor y no afectan el factor de carga.

12. En una tabla de hash con linear probing, si se inserta un elemento y este causa una colisión, podemos afirmar que siempre se cumple que:

- A. El índice donde va a quedar el nuevo elemento es igual al índice asignado + 1.
- B. El índice donde va a quedar el nuevo elemento es mayor al índice asignado al elemento en colisión.
- C. El índice donde va a quedar el nuevo elemento es menor al índice asignado al elemento en colisión.
- D. Ninguna de las otras opciones.

Explicación:

Opción D:

La opción A es falsa porque ese índice podría estar ocupado.

La opción B es falsa porque si el sondeo llega al final del arreglo, reinicia desde el índice cero del arreglo y podría ubicar a la pareja llave-valor en un índice menor al índice originalmente asignado.

La opción C es falsa porque el sondeo podría encontrar un índice libre mayor al originalmente asignado.

13. En una tabla de hash vacía donde cada letra del alfabeto se asocia con un valor numérico ascendente desde 'a'=1 hasta 'z'=26, se utiliza el método de división aplicando la suma de estos valores para determinar la posición de las llaves alfabéticas en una tabla de 100 posiciones. Dada la inserción exitosa de la llave "aciertos", que se asigna a una posición específica de la tabla tras calcular la suma de los valores numéricos de sus letras y aplicar el módulo 100 al resultado, ¿cuál de los siguientes conjuntos de llaves generará colisiones con esa llave en la tabla?

- A. "stop", "pots", "spot".
- B. "cortesía", "teóricas", "sectario".
- C. "amar", "rama", "mara".
- D. Ninguna de las anteriores.

Explicación:

Opción B: Todos son anagramas de la llave: "aciertos".

14. Agregar el conjunto de llaves {"A", "C", "E", "K", "M"} en el orden dado en una Tabla Hash inicialmente con $M = 5$ (capacidad), manejo de colisiones con Linear Probing, factor de carga $N/M = 0.5$ con N número de llaves y política de rehash aproximando al primo mayor al doble del tamaño actual. A continuación, los valores de hash(...) correspondientes a cada llave: $\text{hash}("A") = 65$, $\text{hash}("C") = 67$, $\text{hash}("E") = 69$, $\text{hash}("K") = 75$, $\text{hash}("M") = 77$. Se aplica la función $\text{hashValue}(\text{key}) = \text{abs}(\text{hash}(\text{key})) \% M$ ¿Cuál es el estado final de la Tabla Hash?

Nota: Una entrada con valor guion: -, significa que está vacía.

- A. ["A", "K", "C", "M", "E"] $M=5$
- B. [-, -, -, -, "A", "K", "C", "M", "E"] $M=11$
- C. [-, -, -, -, "A", "K", "C", "M", "E", -] $M=11$
- D. ["M", "C", -, "E", -, -, -, -, "K", "A"] $M=11$

Explicación:

Opción D:

Primer intento:

$M = 5$, factor de carga $N/M = 0.5$ con N número de llaves

Política de rehash aproximando al primo mayor al doble del tamaño actual

Inserción {"A", "C", "E", "K", "M"} = 65, 67, 69, 75, 77

La función $\text{hashValue}(\text{key}) = \text{abs}(\text{hash}(\text{key})) \% M$

$\text{hashValue}(\text{"A"}) = 65 \% 5 = 0$. Factor de carga = $1/5 = 0.2$

$\text{hashValue}(\text{"C"}) = 67 \% 5 = 2$. Factor de carga = $2/5 = 0.4$

$\text{hashValue}(\text{"E"}) = 69 \% 5 = 4$. Factor de carga = $N/M = 3/5 \approx 0.6$, hay rehashing

Intento final:

$M = 11$, factor de carga $N/M = 0.5$ con N número de llaves

Política de rehash aproximando al primo mayor al doble del tamaño actual (11)

Inserción $\{\text{"A"}, \text{"C"}, \text{"E"}, \text{"K"}, \text{"M"}\} = 65, 67, 69, 75, 77$

La función $\text{hashValue}(\text{key}) = \text{abs}(\text{hash}(\text{key})) \% M$

$\text{hashValue}(\text{"A"}) = 65 \% 11 = 10$. Factor de carga = $1/11 \approx 0.09$

$\text{hashValue}(\text{"C"}) = 67 \% 11 = 1$. Factor de carga = $2/11 \approx 0.18$

$\text{hashValue}(\text{"E"}) = 69 \% 11 = 3$. Factor de carga = $3/11 \approx 0.27$

$\text{hashValue}(\text{"K"}) = 75 \% 11 = 9$. Factor de carga = $4/11 \approx 0.3$

$\text{hashValue}(\text{"M"}) = 77 \% 11 = 0$ (se inserta mediante linear probing en el primer espacio disponible, que sería el índice 0, sin conflictos). Factor de carga = $5/11 \approx 0.45$ (nunca se llegó al nuevo límite)

15. Al usar Sondeo Lineal para manejar colisiones en una tabla de hash, ¿cuál de los siguientes es un problema comúnmente asociado exclusivamente con este método, especialmente cuando se utiliza con funciones de hash de baja calidad?

- A. Clustering secundario.
- B. No hay colisiones.
- C. Clustering primario.
- D. Rehashing.

Explicación:

Opción C: El clustering primario ocurre en el sondeo lineal cuando varios elementos colisionan y se agrupan en bloques consecutivos, lo que aumenta las futuras colisiones y dificulta la inserción y búsqueda. Este problema se agrava con funciones de hash de baja calidad (ineficientes).