

Respuestas

a) Cola final : vacia

b) Quien quedo al frente : Nadien, la cola es la vacia

c) Total de personas en la cola : 0

Ejercicio de Ordenamiento y Busqueda

Parte 1. Ordenamiento de Arreglos

Arreglo Inicial

[7, 3, 5, 2]

Recordatorios

Getta pasada compara elementos consecutivos e intercambia si estan en orden incorrectivo $N = 4 \rightarrow$ Se hacen 3 pasadas completas

Pasada 1

Comparacion 1:

7 ? 3 \rightarrow 7 > 3 \rightarrow Se intercambia

Arreglo: [3, 7, 5, 2]

Comparacion 2 $7 \ ? \ 5 \rightarrow 7 > 5 \rightarrow$ Se Intercambian

Arreglo: [3, 5, 7, 2]

Comparacion 3 $7 \ ? \ 2 \rightarrow 7 > 2 \rightarrow$ Se Intercambia

Arreglo: [3, 5, 2, 7]

Estado tras pasada 1:

[3, 5, 2, 7]

Pasada 2Comparacion 1 $3 \ ? \ 5 \rightarrow 3 < 5 \rightarrow$ no hay Cambio

Arreglo: [3, 5, 2, 7]

Comparacion 2 $5 \ ? \ 2 \rightarrow 5 > 2 \rightarrow$ Se Intercambia

Arreglo [3, 2, 5, 7]

Comparacion 3 $5 \ ? \ 7 \rightarrow 5 < 7 \rightarrow$ No hay Cambio

Arreglo: [3, 2, 5, 7]

Estado tras pasada 2:

[3, 2, 5, 7]

Pasada 3Comparacion 1 $3 \ ? \ 2 \rightarrow 3 > 2 \rightarrow$ Se Intercambia

Arreglo: [2, 3, 5, 7]

Comparacion 2 $3 \ ? \ 5 \rightarrow 3 < 5 \rightarrow$ No hay CambioComparacion 3 $5 \ ? \ 7 \rightarrow 5 < 7 \rightarrow$ no hay Cambio

Estado tras pasada 3

[2, 3, 5, 7]

Arreglo Final Ordenado (Burbuja)

[2, 3, 5, 7]

01 Dic 25

Elsy Jocelyn Rodinccz Juarez

Ejercicio 2 Ordenamiento por Selección (Selection Sort)

Arreglo Inicial

[9, 4, 6, 1, 3]

Recordatorio: En cada iteración i se busca el mínimo desde la posición i hasta el final y se intercambia con el elemento en i.

Iteración 1 ($i = 0$)

Subarreglo Considerado: [9, 4, 6, 1, 3]

Buscar mínimo

9 vs 4 \rightarrow mínimo = 4

4 vs 6 \rightarrow mínimo = 4

4 vs 1 \rightarrow mínimo = 1

1 vs 3 \rightarrow mínimo = 1

Mínimo encontrado = 1 (posición 3)

Intercambio:

Intercambiar Posición 0 \leftrightarrow 3

Arreglo

[1, 4, 6, 9, 3]

Intercambios hasta ahora: 1

Iteración 2 ($i = 1$)

Subarreglo: [4, 6, 9, 3]

Buscar mínimo

4 vs 6 \rightarrow mínimo = 4

4 vs 9 \rightarrow mínimo = 4

4 vs 3 \rightarrow mínimo = 3

Mínimo = 3 (Posición 4)

Intercambio

Posición 1 \leftrightarrow 4

Arreglo

[1, 3, 6, 9, 4]

Intercambios acumulados: 2

Interación 3 ($i = 2$)

Subarreglo: [6, 9, 4]

Buscar Mínimo

$$6 \text{ vs } 9 \rightarrow \text{mínimo} = 6$$

$$6 \text{ vs } 4 \rightarrow \text{mínimo} = 4$$

Mínimo = 4 (Posición 4)

Intercambio

Posición 2 \leftrightarrow 4

Arreglo

[1, 3, 4, 9, 6]

Intercambios acumulados: 3

Interación 4 ($i = 3$)

Subarreglo: [9, 6]

Buscar mínimo:

$$9 \text{ vs } 6 \rightarrow \text{mínimo} = 6$$

Mínimo = 6 (Posición 4)

Intercambio

Posición 3 \leftrightarrow 4

Arreglo

[1, 3, 6, 9, 4]

Intercambios acumulados: 4

Interación 5 ($i = 4$)solo queda un elemento \rightarrow no se hace nada

Intercambios no cambian

Arreglo final Ordenado

[1, 3, 4, 6, 9]

Pregunta: ¿Cuántos intercambios se hicieron?

Se realizaron 4 intercambios

Ejercicio 3 Ordenamiento por Inserción (Inserción Sort)

Arreglo Inicial

[8, 2, 5, 1, 4]

Inserción Sort

Interacción 1 ($i=1$)

Clave = 2

Subarreglo ordenado previo: [8]

Comparaciones

1. $2 < 8 \rightarrow$ si \rightarrow se recorre 8 a la derecha

Movimiento

[8, 8, 5, 1, 4]

Insertar clave 2 en posición 0

[2, 8, 5, 1, 4]

Comparaciones acumuladas: 1

Interacción 2 ($i=0$)

Clave = 5

Subarreglo previo ordenado: [2, 8]

Comparaciones

1: $5 < 8 \rightarrow$ si \rightarrow mover 8

[2, 8, 8, 1, 4]

2: $5 < 2 \rightarrow$ no \rightarrow detener

Insertar clave 5 en posición 1

[2, 5, 8, 1, 4]

Comparaciones acumuladas: 3

Interacción 3 ($i=3$)

Clave = 1

Subarreglo previo ordenado: [2, 5, 8]

Comparaciones

1: $1 < 8 \rightarrow$ mover 8

[2, 5, 8, 8, 4]

2: $1 < 5 \rightarrow$ mover 5

[2, 5, 5, 8, 4]

3: $1 < 2 \rightarrow$ mover 2

[2, 2, 5, 8, 9]

4. Llegamos al inicio \rightarrow Insertar clave 4 en posición 0

[1, 2, 5, 8, 9]

Comparación acumuladas +1

Interacción 4 ($i = 4$)

Clave = 4

Subarreglo previo ordenado: [1, 2, 5, 8]

Comparaciones

1. $4 < 8 \rightarrow$ mover 8

[1, 2, 5, 8, 8]

2: $4 < 5 \rightarrow$ mover 5

[1, 2, 5, 5, 8]

3: $4 < 2 \rightarrow$ no detener

Insertar clave 4 en posición 2

Comparaciones acumuladas: +0

Arreglo final Ordenamiento:

[1, 2, 4, 5, 8]

Respuesta Final

¿Cuántas Comparaciones se realizaron?

Se realizaron 10 comparaciones en total

Parte 2: Búsqueda en Arreglos

Ejercicio 5

Arreglo

[4, 9, 2, 7, 5, 8]

Elemento a buscar +1

Proceso de Búsqueda

Nº Comparación Elemento arreglo ¿Es +? Resultado

Comparación 1 4 No Continuar

Comparación 2 9 No Continuar

Comparación 3 2 No Continuar

Comparación 4 7 Sí Elemento encontrado

¿Cuántas comparaciones se realizaron?

Se realizaron 4 comparaciones

Ejercicio 9 - MergeSort - División y Cambiar

Arreglo inicial

[10, 7, 3, 8, 2, 6]

1) División recursiva (árbol de particiones)

[10, 7, 3, 8, 2, 6]

[10, 7, 3]

'
[8, 2, 6]

[10] [7, 3]

' '
[8] [2, 6]

[7] [3]

' '
[2] [6]

Se divide hasta subarreglos de [2] [6]

tamaño 1

2) Proceso de Combinación (marge)

A) Mezcla en el subárbol izquierdo

- Marge [7] y [3] → Comparar 7 y 3 → resultado [3, 7]
- Marge [10] y [3, 7] → Pássos
- Comparar 10 y 3 + tomar 3
- Comparar 10 y 4 → tomar 4
- quedó 10 → tomar 10

Resultado [3, 7, 10]

B) Mezclas en el subárbol derecho

- Marge [2] y [6] → Comparar 2 y 6 + resultado [2, 6]
- Marge [8] y [7, 6] → tomar 2
- Comparar 8 y 2 + tomar 2
- Comparar 8 y 6 + tomar 6
- quedó 8 → tomar 8

Resultado [2, 6, 8]

C) Marge final (contar las dos mitades)

• Marge $[3, 7, 10]$ (izquierdo) con $[2, 6, 8]$ (derecha)

• Comparar $3 \text{ vs } 2 \rightarrow$ tomar 2 \rightarrow Salida $[2]$

• Comparar $3 \text{ vs } 6 \rightarrow$ tomar 3 \rightarrow Salida $[2, 3]$

• Comparar $7 \text{ vs } 6 \rightarrow$ tomar 6 \rightarrow Salida $[2, 3, 6]$

• Comparar $7 \text{ vs } 8 \rightarrow$ tomar 7 \rightarrow Salida $[2, 3, 6, 7]$

• Comparar $10 \text{ vs } 8 \rightarrow$ tomar 8 \rightarrow Salida $[2, 3, 6, 7, 8]$

• derecha vacia \rightarrow volver resto de izquierdo (10) \rightarrow Salida
 $[2, 3, 6, 7, 8, 10]$

Resultado final despues de mezclar

$[2, 3, 6, 7, 8, 10]$

Parte 2 - Ejercicio 6 - Búsqueda Binaria

Arreglo ordenado

$[3, 6, 8, 12, 15, 18, 20]$

índices

0	1	2	3	4	5	6
3	6	8	12	15	18	20

valor a buscar: 15

Paso 1

• Inicio = 0

• fin = 6

• media = $(0+6)/2 = 3$

Elemento central

arr $[3] = 12$

Comparacion

$12 < 15$ el numero buscado esta en la mitad derecha

Eliminar: la mitad izquierda $[3, 6, 8, 12]$

Nuevo rango: índices 4 a 6

Paso 2

• Inicio = 4

• fin = 6

• media = $(4+6)/2 = 5$

Elemento Central

$$\text{arr}[S] = 18$$

Comparación

$18 > 15 \rightarrow$ el número buscado está en la mitad izquierda

Eliminar : la mitad derecha [18, 20]

Nuevo rango : índices 4 a 4

Paso 3

- $\text{Inicio} = 4$
- $\text{Fin} = 4$
- $\text{medio} = (4+4)/2 = 4$

Elemento Central

$$\text{arr}[4] = 15$$

Comparación:

$15 == 15 \rightarrow$ Elemento encontrado

Respuesta

El elemento 15 se encuentra en el paso 3

Ejercicio 4 - Búsqueda Binaria (elemento No existente)

Arreglo (ordenado) y sus índices:

índices : 0, 1, 2, 3, 4, 5

Valores : 2, 5, 9, 14, 21, 30

Elemento a buscar : 10

Interacción 1

- $\text{Inicio} = 0, \text{Fin} = 5$
- $\text{medio} = (0+5)/2 = 2 \rightarrow \text{arr}[2] = 9$
- Comparación : $9 < 10 \rightarrow$ descartar la mitad izquierda (incluyendo media)
- Nuevo intervalo : $\text{Inicio} = 3, \text{Fin} = 5$

Interacción 2

- $\text{Inicio} = 3, \text{Fin} = 5$
- $\text{medio} = (3+5)/2 = 4 \rightarrow \text{arr}[4] = 21$
- Comparación : $21 > 10 \rightarrow$ descartar la mitad derecha (incluyendo medio)
- Nuevo Intervalo : $\text{Início} = 3, \text{Fin} = 3 (\text{Fin} = \text{medio} - 1 = 3)$

Interación 3

Inicio = 3, fin = 3

• media = $\lfloor \frac{3+5}{2} \rfloor = 4 \rightarrow arr[4] = 21$

Comparacion $14 > 10 \rightarrow$ desartar la mitad derecha (incluyendo medio)

• Nuevo Intervalo: Inicio = 3, fin = 2 ($fin = media - 1 = 2$)

Termina el algoritmo

Inicio 3 y fin = 2 es decir inicio > fin

Eso significa que el intervalo de búsqueda queda vacío \rightarrow el elemento no está en el arreglo

• Comparaciones realizadas 3

• Resultado 10 no encontrado

¿Por qué Condición termina el algoritmo?

Por que la condición de parada es que el intervalo de búsqueda se vuelva vacío. En la implementación habitual esto se detecta cuando $inicio > fin$. En ese momento no quedan posiciones por inspeccionar

Parte 3 Ejercicios Integradores - Ejemplo 8

Arreglo Original

[12, 3, 19, 5, 8, 7]

1: Ordenamiento - Bubble Sort

Pasada 1

1. 12 vs 3 \rightarrow Intercambio $\rightarrow [3, 12, 19, 5, 8, 7]$

2. 12 vs 19 \rightarrow OK

3. 19 vs 5 \rightarrow Intercambio $\rightarrow [3, 12, 5, 19, 8, 7]$

4. 19 vs 8 \rightarrow Intercambio $\rightarrow [3, 12, 5, 8, 19, 7]$

5. 19 vs 7 \rightarrow Intercambio $\rightarrow [3, 12, 5, 8, 7, 19]$

Estado tras Pasada 1

[3, 12, 5, 8, 7, 19]

Elsy Joselyn Cadiz Juarez

01 DIC 25

Pasada 2

1: 3 vs 12 → On

2: 12 vs 5 → Intercambio → [3, 5, 12, 8, 7, 19]

3: 12 vs 8 → Intercambio → [3, 5, 8, 12, 7, 19]

4: 12 vs 7 → Intercambio → [3, 5, 8, 7, 12, 19]

Estado tras pasada 2:

[3, 5, 8, 7, 12, 19]

Pasada 3

3 vs 5 → On

5 vs 8 → On

8 vs 7 → Intercambio → [3, 5, 7, 8, 12, 19]

Estado tras pasada 3:

[3, 5, 7, 8, 12, 19]

Pasada 4

1: 3 vs 5 → On

2: 5 vs 7 → Intercambio → [3, 7, 5, 8, 12, 19]

Estado tras pasada 4

[3, 7, 5, 8, 12, 19]

Pasada 5

1: 3 vs 7 → Intercambio → [1, 3, 5, 8, 12, 19]

Estado tras pasada 5:

[1, 3, 5, 8, 12, 19]

Arreglo Final ordenado

[1, 3, 5, 8, 12, 19]

2. Búsqueda Binaria del número 8

Arreglo Ordenado

[1, 3, 5, 8, 12, 19]

índices

0 1 2 3 4 5

1, 3, 5, 8, 12, 19

valor a buscar: 8

Iteración 1

Inicio = 0

fin = 5

$$\text{medio} = (0+5)/2 = 2 + \text{arr}[2] = 5$$

5 < 8 \rightarrow buscar en la derecha

nuevo intervalo inicio = 3 fin = 5

Iteración 2

Inicio = 3

fin = 5

$$\text{medio} = (3+5)/2 = 4 \rightarrow \text{arr}[4] = 12$$

12 > 8 \rightarrow buscar en la izquierda

nuevo intervalo: inicio = 3, fin = 3

Iteración 3

Inicio = 3

fin = 3

$$\text{medio} = (3+3)/2 = 3 \rightarrow \text{arr}[3] = 8$$

Encontrado

Resultados finales

Arreglo ordenado,

[1, 3, 5, 8, 12, 19]

Búsqueda binaria

el número 8 se encontró en la
iteración 3.

Ejercicio 9: Elegir el Mejor Algoritmo

Situación: arreglo escritamente inverso

[5, 4, 3, 2, 1] (peor caso para muchos algoritmos simples)

Comparación rápida de candidatos

► Bubble Sort

- Tiempo: $O(n^2)$ en el peor caso

- Para un arreglo inverso realiza la máxima numero de comparaciones y el maximo numero de intercambios (muy lento para n grande)

- No recomendable aquí

► Insertion Sort

- Tiempo: $O(n^2)$ en el peor caso

- Es muy bueno cuando el arreglo está casi ordenado pero para un arreglo inverso hace el maximo de desplazamientos (cada nueva clave debe moverse hasta el inicio)

- No es suficiente

► Selection Sort

- Tiempo: $O(n^2)$ en todas las entradas (siempre hace $-n^2/2$ comparaciones)

- Número de intercambios es bajo ($n - 1$) pero las comparaciones siguen siendo numerosas. Para n pequeño puede estar bien pero asintóticamente es peor que $O(n \log n)$.

► Merge Sort

- Tiempo: $O(n \log n)$ garantizado para cualquier entrada (mejor comportamiento asintótico, no depende del orden inicial)

- Necesita espacio adicional $O(n)$ para mezclar, pero para rendimiento (asintótico no depende del orden inicial) es la opción más segura para entradas "mal ordenadas" como ésta

Conclusion razonad en parejas

Es el arreglo es completamente inverso, lo que convierte a Bubble e insertion en sus peores casos - ambos $O(n^2)$ con muchas intercambios / desplazamiento

- Selection tambien hace $O(n^2)$ comparaciones (aunque pocos swaps) asi que no es asintoticamente mejor
- Quick - Puede ser excelente si eliges pivote bien corre riesgo de caer en $O(n^2)$ con este arreglo
- Merge Sort ofrece $O(\log n)$ garantizando independientemente del orden de entrada, pero eso es la mejor elección segura para este caso concreto.

1) Bubble Sort versión optimizada - inner loop reduce cada pasada

Algoritmo: para $i = 0 \dots n-2$ comparar $j = 0 \dots n-i-1$ y swap
 Si $a[j] > a[j+1]$

Arreglo inicial $[5, 4, 3, 2, 1]$

• Pasada 1 ($i=0$) - $j = 0..3$ (4 comparaciones)

Comps: 4 swaps: 9

Transformaciones:

$[5, 4, 3, 2, 1] \rightarrow [4, 5, 3, 2, 1] + [4, 3, 5, 2, 1] \rightarrow [4, 3, 2, 5, 1] + [4, 3, 2, 1, 5]$

• Pasada 2 ($i=1$) - $j = 0..2$ (3 comparaciones)

Comps: $+2 = 7$, swaps: $+3 = 7$

$\rightarrow [3, 2, 1, 4, 5]$ (interminados mostrando arriba)

• Pasada 3 ($i=2$) - $j = 0..1$ (2 comparaciones)

Comps: $+2 = 9$, swaps: $+2 = 9$

$\rightarrow [2, 1, 3, 4, 5]$

Pasada 4 ($i=3$) - $j = 0..0$ (1 comparación)

Comps: $+1 = 10$, swaps: $+1 = 10$

$\rightarrow [1, 2, 3, 4, 5]$

Totales

• Comparacion = $10 \cdot (n-1)/2$

• Swaps = 10 (en el caso resuelto cada comparación produce un swap)

Ejercicio 10 Simulación de Quicksort

Primera Partición (Pivote = 9)

Arreglo Inicial

- Menores que 9:

[4, 7, 3, 1]

- Iguales (Pivote):

[9]

Mayores que 9

[10]

Resultado:

[4, 7, 3, 1] + [9] + [10]

2: Quicksort

2: Quicksort a [4, 7, 3, 1]

Pivote = 4

Participación:

Menores: [3, 1]

Igual [4]

Mayores [7]

Resultado Parcial

[3, 1] + [9] + [7]

Ahora ordena [3, 1]

3. Quicksort a [3, 1]

Pivote = 3

Partición:

Menores [1]

Igual [3]

Mayores []

Resultado [1] + [3]

No hay más llamadas; ambas son simples

Resultado de ordenar menores de pivote 9

Partes Combinadas

[1, 3] + [4] + [7]

[1, 3, 4, 7]

01 DIC 25

Elsy Joselyn Godinez chavez

El subarreglo de mayores de A era

$[10] \rightarrow$ ya esta ordenado

Resultado Final

Se combina todas partes $[1, 3, 4, 7] + [9] + [10]$

Arreglo final Ordenado

$[1, 3, 4, 7, 9, 10]$