

Algoritmos y Programación II – Cátedra Lic. Gustavo Carolo

Evaluación Final Resuelta – 2008-07-21

Teoría Tema 1

Dado el siguiente lote de datos:

398, 117, 18, 287, 219⁺, 48, 64, 367⁺, 55⁺, 314, 378⁺, 150, 15, 159, 34, 161⁺, 78, 125⁺, 99, 214, 324⁺

Se pide:

- a) Mostrar cómo queda el B-tree de $m=2$ resultante de ingresar los datos en el orden dado. Mostrar los resultados parciales luego de insertar un elemento marcado con ⁺. Mostrar como queda el B-tree obtenido tras eliminar los siguientes elementos en el orden dado:

159, 150, 214

En este caso no es necesario mostrar el árbol para cada eliminación.

- b) Desarrollar los algoritmos en orden ascendente de Radix sort y de Quick sort hasta haber entrado en tres niveles de recursividad utilizando el primer elemento como pivote.
- c) Desarrollar el algoritmo de selección de natural con un buffers de tamaño 4.
- d) Desarrollar las dos primeras fases del algoritmo de merge polifásico para 3 vías de entrada con 21 particiones ordenadas tomadas del lote de datos.
- e) Si hizo correctamente el desarrollo del Quick sort del punto b habrá particionado con 4 pivotes, sin embargo para tres niveles de recursividad se esperarían 7 pivotes (1 para el primero, 2 para el segundo y 4 para el tercero). ¿por qué no se dio ese caso? ¿cuál habría sido el peor caso con estos 21 elementos? ¿cuántas entradas recursivas habríamos necesitado para ordenar el peor caso? ¿y para el mejor caso? Responda cada pregunta en no más de dos renglones.

a) Ejercicio de B-trees

Insertar: 398, 117, 18, 287, 219, 48, 64, 367, 55, 314, 378, 150, 15, 159, 34, 161, 78, 125, 99, 214, 324
Luego eliminar el 159, el 150 y finalmente el 214

Resolucion:

Primero inserto 5 elementos hasta obtener una raiz: **398, 117, 18, 287, 219**

(219)

(18 | 117) (287 | 398)

Luego inserto: **48, 64, 367**

(219)

(18 | 48 | 64 | 117) (287 | 367 | 398)

Insertando: **55** obtengo mi segundo elemento de la raiz

(55 | 219)

(18 | 48) (64 | 117) (287 | 367 | 398)

E insertando: **314, 378** logro un tercer elemento en la raiz

(55 | 219 | 367)

(18 | 48) (64 | 117) (287 | 314) (378 | 398)

Llegamos al cuarto elemento de la raiz insertando: **150, 15, 159, 34, 161**

(55 | 150 | 219 | 367)

(15 | 18 | 34 | 48) (64 | 117) (159 | 161) (287 | 314) (378 | 398)

Completamos un poco más con: **78, 125**

(55 | 150 | 219 | 367)

(15 | 18 | 34 | 48) (64 | 78 | 117 | 125) (159 | 161) (287 | 314) (378 | 398)

Y finalizamos el arbol obteniendo un nivel mas con: **99, 214, 324**

raiz en 150 y dos subarboles de

(55 | 99)

(15 | 18 | 34 | 48) (64 | 78) (117 | 125)

y

(219 | 367)

(159 | 161 | 214) (287 | 314 | 324) (378 | 398)

Ahora debemos eliminar: **159, 150, 214.**

Eliminar 159 no tiene complicacion. Eliminamos 150 y obtenemos.

(55 | 99 | 219 | 367)

(15 | 18 | 34 | 48) (64 | 78) (117 | 125 | 161 | 214) (287 | 314 | 324) (378 | 398)

y sin el 214 obtenemos el resultado final

(55 | 99 | 219 | 367)

(15 | 18 | 34 | 48) (64 | 78) (117 | 125 | 161) (287 | 314 | 324) (378 | 398)

b) Radix sort

Ascendente:

Paso 0 Paso 1 Paso 2 Paso 3

398	150	314	015
117	161	214	018
018	064	015	034
287	314	117	048
219	034	018	055
048	214	219	064
064	324	324	078
367	055	125	099
055	015	034	117
314	125	048	125
378	117	150	150
150	287	055	159

015	367	159	161
159	398	161	214
034	018	064	219
161	048	367	287
078	378	378	314
125	078	078	324
099	219	287	367
214	159	398	378
324	099	099	398

c) Selección Natural

Set de Datos: 398, 117, 18, 287, 219, 48, 64, 367, 55, 314, 378, 150, 15, 159, 34, 161, 78, 125, 99, 214, 324

Buffer primario: XX XX XX XX
 Buffer auxiliar: XX XX XX XX
 No sale nada
 Entra 398

Buffer primario: 398 117 219 287
 Buffer auxiliar: XX XX XX XX
 Sale 117
 Entra 48

Buffer primario: 398 XX XX XX
 Buffer auxiliar: XX XX XX XX
 No sale nada
 Entra 117

Buffer primario: 398 XX 219 287
 Buffer auxiliar: **48** XX XX XX
 No sale nada
 Entra 64

Buffer primario: 398 117 XX XX
 Buffer auxiliar: XX XX XX XX
 No sale nada
 Entra 18

Buffer primario: 398 XX 219 287
 Buffer auxiliar: **48 64** XX XX
 No sale nada
 Entra 367

Buffer primario: 398 117 18 XX
 Buffer auxiliar: XX XX XX XX
 No sale nada
 Entra 287

Buffer primario: 398 367 219 287
 Buffer auxiliar: **48 64** XX XX
 Sale 219
 Entra 55

Buffer primario: 398 117 18 287
 Buffer auxiliar: XX XX XX XX
 Sale 18
 Entra 219

Buffer primario: 398 367 XX 287
 Buffer auxiliar: **48 64 55** XX
 No sale nada
 Entra 314

Buffer primario: 398 367 314 287
Buffer auxiliar: **48 64 55 XX**
Sale 287
Entra 378

Buffer primario: 398 367 314 378
Buffer auxiliar: **48 64 55 XX**
Sale 314
Entra 150

Buffer primario: 398 367 XX 378
Buffer auxiliar: **48 64 55 150**
Sale 367
No entra nada

Buffer primario: 398 XX XX 378
Buffer auxiliar: **48 64 55 150**
Sale 378
No entra nada

Buffer primario: 398 XX XX XX
Buffer auxiliar: **48 64 55 150**
Sale 398
Intercambio de buffers!
No entra nada

Buffer primario: **48 64 55 150**
Buffer auxiliar: XX XX XX XX
Nueva particion por recién intercambiado!
Se cierra la particion p1: 18, 117, 219, 287, 314, 367, 378, 398
Comienza saliendo 48
Entra 15

Buffer primario: XX 64 55 150
Buffer auxiliar: **15 XX XX XX**
No sale nada
Entra 159

Buffer primario: 159 64 55 150
Buffer auxiliar: **15 XX XX XX**
Sale 55
Entra 34

Buffer primario: 159 64 XX 150
Buffer auxiliar: **15 34 XX XX**
No sale nada
Entra 161

Buffer primario: 159 64 161 150
Buffer auxiliar: **15 34 XX XX**
Sale 64
Entra 78

Buffer primario: 159 78 161 150
Buffer auxiliar: **15 34 XX XX**
Sale 78
Entra 125

Buffer primario: 159 125 161 150
Buffer auxiliar: **15 34 XX XX**
Sale 125
Entra 99

Buffer primario: 159 XX 161 150
Buffer auxiliar: **15 34 99 XX**
No sale nada
Entra 214

Buffer primario: 159 214 161 150
Buffer auxiliar: **15 34 99 XX**
Sale 150
Entra 324

Buffer primario: 159 214 161 324
Buffer auxiliar: **15 34 99 XX**
Sale 159
No entra nada

Buffer primario: XX 214 161 324

Buffer auxiliar: **15 34 99** XX

Sale 161

No entra nada

Buffer primario: XX 214 XX 324

Buffer auxiliar: **15 34 99** XX

Sale 214

No entra nada

Buffer primario: XX XX XX 324

Buffer auxiliar: **15 34 99** XX

Sale 324

No entra nada

Salida final:

p1: 18, 117, 219, 287, 314, 367, 378, 398

p2: 48, 55, 64, 78, 125, 150, 159, 161, 214, 324

d) Merge Polifásico

Via 1	Via 2	Via 3	Via 4
D	D	D	D
D	D	D	D
D	D	D	D
D	398	117	117,398
18	287	219	18,219,287
48	64	367	48,64,367
55	314	378	55,314,378
Cierro Via 4			
150	15	D	15,150
159	34	D	34,159
161	78	D	78,161
125	99	117,398	99,117,125,398
Cierro Via 4			

b) Qsort

Paso 1	398	117	18	287	219	48	64	367	55	314	378	150	15	159	34	161	78	125	99	214	324
Paso 2	324	117	18	287	219	48	64	367	55	314	378	150	15	159	34	161	78	125	99	214	398
Paso 3	125	117	18	287	219	48	64	214	55	314	99	150	15	159	34	161	78	324	378	367	398
Paso 4	99	117	18	78	34	48	64	15	55	125	314	150	214	159	219	161	287	324	367	378	398

e) Preguntas teóricas

¿por qué no se dio ese caso?

Porque el pivote resultó ser el mayor y entonces no hubo intercambios y quedó una partición vacía.

¿cuál habría sido el peor caso con estos 21 elementos?

El arreglo ordenado descendentemente, porque esto pasaría siempre.

¿cuántas entradas recursivas habríamos necesitado para ordenar el peor caso?

21, una por cada elemento

¿y para el mejor caso?

Si el pivote elegido resulta siempre en la mitad del arreglo, se podría ordenar 16 elementos en 4 pasos y 32 elementos en 5 pasos. En nuestro caso 21 elementos necesitarían 5 pasos también.