

Guía Precontrol 1

Profesores: Nelson Baloian, Patricio Poblete
Benjamín Bustos
Sebastián Ferrada

Auxiliares: Vicente Olivares, Ricardo Valdivia
Sebastián Acuña
Valentina Aravena

P1. P1 Control Otoño 2023

Suponga que se dispone de una función `sgte_dato()` que cada vez que es llamada entrega el siguiente dato de una secuencia de entrada cuya longitud no se conoce a priori. Esos datos son números enteros y la secuencia termina cuando aparece un cero.

Usted debe escribir un trozo de programa que haga uso de esta función para obtener estos datos, y que luego imprima primero todos los números negativos, en el orden inverso al orden en que aparecieron, seguidos de los números positivos, en el orden en que aparecieron.

Para lograr esto, usted solo puede usar una pila y una cola. *Ninguna otra estructura de datos está permitida.*

```
1 Ejemplo:  
2 Entrada: 15, -10, 7, -5, -8, 6, 14, 0  
3 Salida: -8, -5, -10, 15, 7, 6, 14
```

P2. Ecuaciones de Recurrencia

Resuelva las siguientes ecuaciones de recurrencia

a)

$$a_n = 3a_{n-1} + 4a_{n-2}$$

$$a_0 = 2$$

$$a_1 = 3$$

b)

$$f(n) = 2f(\sqrt{n}) + \log_2 n$$

Sugerencia: Haga el cambio de variable $n = 2^k$ y aplique el Teorema Maestro

c)

$$a_n = 2a_{n-1} + a_{n-2}$$

$$a_0 = 0$$

$$a_1 = 4$$

P3. P2.2 Control Otoño 2023

Considere la siguiente ecuación de recurrencia:

$$T(n) = 8T(n-1) - 15T(n-2)$$
$$T(0) = 1, T(1) = 1$$

- Escriba una función recursiva que calcule $T(n)$ en base a esta ecuación e indique el orden de magnitud del tiempo que demoraría (no es necesario ser preciso, basta que diga si es lineal, cuadrático, logarítmico, exponencial, etc.)
- A continuación, use una estrategia de tabulación para obtener un algoritmo más eficiente. Indique el orden de magnitud del tiempo que demoraría.
- Finalmente, resuelva analíticamente la ecuación para $T(n)$. Escriba su solución al reverso de esta página.

P4. Lista Rotada

Supongamos que tenemos una lista ordenada de un cierto largo n , por ejemplo:

[13, 20, 34, 41, 55, 62, 75, 84, 93]

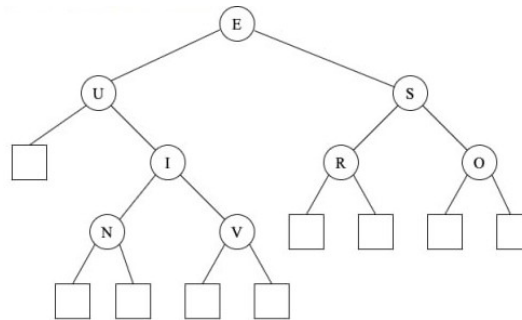
Si la desplazamos circularmente una cierta cantidad de posiciones hacia la derecha, lo que resulta se llama una lista rotada. Por ejemplo,

[75, 84, 93, 13, 20, 34, 41, 55, 62]

Escriba una función `encuentra_minimo(a)` que reciba como parámetro un arreglo `a` conteniendo una lista rotada, y que retorne el subíndice en donde se encuentra el mínimo elemento de la lista. Su función debe operar en tiempo $\Theta(\log n)$. Justifique por qué su algoritmo cumple con este requerimiento.

P5. Árbol de letras

Para el siguiente árbol:



- Calcule su altura.
- Indique en qué orden se visitarían los nodos según el tipo de recorrido.
 - Preorden:
 - Inorden:
 - Postorden:

P6. Buscar elementos en arreglos ordenados. Suponga que se dispone de un arreglo A de capacidad ilimitada (es decir, $\text{len}(A) = \infty$) que contiene números enteros y en donde los índices del arreglo comienzan en 1. Los primeros n casilleros del arreglo (n es un valor desconocido) contienen números enteros en orden ascendente, y a partir del casillero $n + 1$ en adelante se almacena el valor ∞ . Se propone el siguiente algoritmo para buscar el índice del arreglo donde se almacena un cierto valor x :

- Se verifica si el primer casillero del arreglo (índice = 1) contiene x . Si lo contiene, retornar dicho índice.
- Continuar revisando casilleros de la siguiente manera: si en una iteración se revisa el casillero de índice i , en la siguiente se debe revisar el casillero de índice $2i$ (esto es: $i = 1, 2, 4, 8, 16, \dots$). Repetir mientras el número almacenado en el casillero de índice i sea menor que x .
- Si en el casillero de índice i se encuentra un número mayor que x , hacer búsqueda binaria entre los casilleros $[i/2, i]$.
- Si en el casillero de índice i se encuentra almacenado ∞ , hacer búsqueda binaria entre los casilleros $[i/2, i]$.

¿Cuál es la complejidad temporal en el peor caso de este algoritmo de búsqueda en función de n ? Fundamente su respuesta.

P7. Matriz ascendente

Suponga que se tiene una matriz (tabla rectangular) a de $m \times n$, en que cada fila y cada columna están ordenadas ascendentemente. Por ejemplo:

$$\begin{pmatrix} 12 & 28 & 35 & 56 & 72 \\ 25 & 33 & 40 & 61 & 80 \\ 37 & 44 & 52 & 65 & 84 \\ 50 & 60 & 71 & 86 & 90 \end{pmatrix}$$

En esta matriz se desea buscar un número dado x

- Suponga que se hace una búsqueda secuencial. ¿Cuánto demora el algoritmo en el peor caso?
- Suponga que se hace búsqueda binaria en cada fila. ¿Cuánto demora en el peor caso?
- Considere un nuevo algoritmo que comienza comparando x contra el elemento de la esquina inferior izquierda ($a_{m,1}$). Indique cuáles elementos podría descartar si $x < a_{m,1}$ y cuáles podría descartar si $x > a_{m,1}$. Si se itera esta idea, ¿cuánto demoraría el algoritmo en el peor caso?

P8. Filtrando números positivos

Suponga que se tiene una lista enlazada a con cabecera, que contiene números enteros. Escriba una función que se pueda invocar como “ $a.filtrapositivos()$ ”, que al ejecutarse elimine de la lista todos los nodos que contienen números ≤ 0 y que deje solo los que contienen números positivos. Utilice las definiciones de Nodo y de Lista que aparecen en los apuntes.