Universidad del Valle

Segundo examen parcial - Fundamentos de análisis y diseño de algoritmos

Duración: 2 horas

Carlos Andres Delgado S, Ing *

17 de Junio 2017

Nombre:_____
Código:_____

1. Estructuras de datos [30 puntos]

1. **(10 puntos)** En el procedimiento **Build-Heap**, se construye el montículo, en un ciclo desde length[A]/2 hasta 1 de manera creciente. Esto es un capricho, pues de igual forma, el montículo puede ser construido, si el ciclo fuese desde 1 hasta $lenght[A]/2$, aplicando Heapify en cada iteración. Indique si es verdadero o falso, justifique su afirmación.

2. **(10 puntos)** Si se deseara que el algoritmo HeapSoft ordenara descendentemente, es suficiente con la siguiente modificación.

```
HeapSoftDesc(A)
    Build-Heap(A)
    for i=2 to n-1
            Heapify(A,i)
```

   Indique si es verdadero o falso, justifique su afirmación.

3. **(10 puntos)** ¿Cómo se puede implementar una cola utilizando pilas? Analice las complejidades de las operaciones.

2. Programación dinámica y voraz [70 puntos]

2.1. Problema del viaje más barato

Sobre el río Cauca hay n embarcaderos. En cada uno de ellos se puede alquilar un bote que permite ir a cualquier otro embarcadero río abajo (es imposible ir río arriba). Existe una tabla de tarifas que indica el coste del viaje del embarcadero i al j para cualquier embarcadero de partida i y cualquier embarcadero de llegada j más abajo en el río $i < j$. Puede suceder que un viaje de i a j sea más caro que una sucesión de viajes más cortos, en cuyo caso se tomaría un primer bote hasta un embarcadero k y un segundo bote para continuar a partir de k. No hay coste adicional por cambiar de bote. A continuación un ejemplo con 4 embarcaderos, se quiere ir de 1 a 4.

	2	**3**	**4**
1	10	40	100
2	-	20	80
3	-	-	5

La solución óptima en este caso es tomar 1 a 2 (costo 10), 2 a 3 (costo 20), 3 a 4 (Costo 5) para un costo total de 35.

2.2. Problema del cambio de monedas

Usted se encuentra administrando un pequeño negocio ubicado en la ciudad de Tulúa. Las personas le compran productos y al momento de pagar usted entrega una devuelta en monedas. Se desea encontrar un algoritmo que minimice el número de monedas que se retornan. El problema se puede describir así:

- Tiene un valor numérico entero a devolver A
- Tiene un conjunto de monedas $B = \{b_1, b_2, ..., b_n\}$
- Se busca encontrar el subconjunto de monedas $B_s = \{b_i, .., b_j, ..., b_n\}$ de tal forma su suma sea igual A y el tamaño de B_s sea el menor posible.

Ejemplo:

- $A = 50$
- $B = \{10, 10, 20, 30, 10, 20\}$
- La solución óptima a este problema es $B_s = \{20, 30\}$ ya que es el subconjunto de menor tamaño que suma 50

2.3. Preguntas

Para cada uno de los problemas anteriores responda:

1. **(4 puntos)** ¿Cual es la complejidad de la solución ingenua? Justifique.

*carlos.andres.delgado@correounivalle.edu.co

2. Cocinemos una deliciosa solución dinámica:

   a) **(9 puntos) Paso 1:** Caracterice la solución optima. Identifique si el problema se puede solucionar mediante divide y vencerás. Observe si existe solapamiento de problemas y sí una solución optima esta compuesta por soluciones óptimas de los subproblemas.

   b) **(7 puntos) Paso 2:** Defina recursivamente el valor de una solución óptima. Identifique la estructura de memorización, que significa cada una de sus posiciones y cómo se calcula cada una. Justifique formalmente.

   c) **(7 puntos) Pasos 3 y 4:** Muestre cómo se calcula el valor de la solución optima de forma Bottom-up.

3. **(8 puntos)** A partir de la caracterización realizada diseñe una solución voraz al problema. Justifique formalmente.

¡Exitos!