Universidad del Valle

Segundo examen opcional
Fundamentos de análisis y diseño de algoritmos
Duración: 2 horas

Carlos Andres Delgado S, Ing *

14 de Junio 2017

Nombre:__________________________________
Código:_________________

1. Estructuras de datos [20 puntos]

Un palíndromo es una cadena de caracteres que se puede leer igual de izquierda a derecha y viceversa (ignorando los espacios entre las palabras). Escriba una función que reciba una cadena almacenada en una lista simplemente enlazada (un solo carácter por cada nodo), y que retorne verdadero o falso, según si la cadena recibida es o no un palíndromo respectivamente.
NOTA: Sólo serán válidas las soluciones que únicamente utilicen las estructuras vistas en clase.

1. **(5 puntos)** ¿Cual es la complejidad de la solución ingenua? Sustente con un ejemplo
2. **(15 puntos)** Diseñe una solución a este problema. Explique porque utilizó las estructuras de datos seleccionadas y de acuerdo a las operaciones realizadas indique y justifique claramente la complejidad del algoritmo diseñado.

2. Ordenamiento [30 puntos]

Para cada uno de los siguientes problemas, describa un algoritmo que encuentre los números solicitados en el tiempo previsto. Para que sus respuestas sean breves, utilice si lo necesita cualquiera de los algoritmos vistos en clase.

1. **(7.5 puntos)** Sea S un arreglo de n números enteros. Encuentre la pareja $x, y \in \mathcal{S}$ que maximice $|x-y|$. Su algoritmo debe correr en tiempo $\mathcal{O}(n)$ en el peor caso.
2. **(7.5 puntos)** Sea S un arreglo **ordenado** de n números enteros. Encuentre la pareja $x, y \in \mathcal{S}$ que maximice $|x-y|$. Su algoritmo debe correr en tiempo $\mathcal{O}(1)$ en el peor caso.
3. **(7.5 puntos)** Sea S un arreglo de n números enteros. Encuentre la pareja $x, y \in \mathcal{S}$ que minimice $|x-y|, x \neq y$. Su algoritmo debe correr en tiempo $\mathcal{O}(n \lg n)$ en el peor caso.
4. **(7.5 puntos)** Sea S un arreglo **ordenado** de n números enteros. Encuentre la pareja $x, y \in \mathcal{S}$ que minimice $|x-y|, x \neq y$. Su algoritmo debe correr en tiempo $\mathcal{O}(n)$ en el peor caso.

3. Programación dinámica y voraz [50 puntos]

3.1. 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.

Inicio	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 30), 3 a 4 (Costo 5).

1. **(5 puntos)** ¿Cual es la complejidad de la solución ingenua? Justifique con un ejemplo:
2. Apliquemos la receta:
   a) **(15 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) **(15 puntos) Paso 2:** Defina recursivamente el valor de una solución óptima. Identifique la estructura de memorización, que significa

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

cada una de sus posiciones y cómo se calcula cada una.

c) **(7.5 puntos) Paso 3:** Calcule el valor de la solución optima de forma Bottom-up. Primero problemas triviales hasta problema general. Explique, puede usar el ejemplo anterior.

d) **(7.5 puntos) Paso 4:** Construya la solución optima a partir de la información adquirida. Identifique desde la solución óptima las decisiones que se tomaron y muestre la salida. Explique, puede usar el ejemplo anterior.

¡Exitos!