

CC4102 - Examen

Prof. Gonzalo Navarro

14 de Diciembre de 2015

P1 (3.0 pt)

Como guía, las preguntas no deberían responderse en más de 5 líneas cada una.

1. ¿Qué significan y cómo se relacionan la complejidad de un problema, una cota inferior a un problema, una cota superior a un problema, la complejidad de de mejor caso de un algoritmo y la complejidad de peor caso de un algoritmo?
2. Describa el modelo de costo usado frecuentemente (y en el curso) para evaluar algoritmos en memoria secundaria. ¿Qué ventajas y desventajas tiene?
3. ¿En qué difieren los conceptos de tiempo promedio, tiempo esperado, y tiempo amortizado?
4. ¿En qué difieren los conceptos de algoritmo probabilístico, aleatorizado, y aproximado?
5. ¿Cómo se relaciona la dificultad de los problemas de decisión con los problemas asociados de optimización?
6. ¿Cómo se refinan los problemas NP-completos en términos de su aproximabilidad? ¿Cómo puede mostrarse que ciertos problemas no son aproximables?

P2 (1.0 pt)

Se desea una estructura de datos para almacenar un conjunto de strings en memoria secundaria. Estos strings pueden ser bastante largos, incluso más que el espacio de una página de disco.

1. Diseñe una variante directa del B-tree para almacenar estos strings: describa la estructura de datos y cómo se busca (no necesita considerar los algoritmos de inserción y borrado). Analice el costo de búsqueda en el modelo de memoria secundaria e indique qué problema tiene su estructura comparada con un B-tree de números.
2. Considere ahora una variante donde en cada nodo del B-tree se guarda un árbol Patricia de las claves en vez de las claves directamente. Nuevamente describa la estructura y el algoritmo de búsqueda, y analice el costo de búsqueda. ¿Cómo se compara el costo con el de un B-tree de números?

P3 (1.0 pt)

Tiene usted $k\$$ y quiere gastarse la mayor cantidad posible en una feria a lo largo de una calle de un solo sentido (es decir, usted solo puede hacer una pasada por ella), de modo que cuando compra algo ya no puede devolverlo (además hay un solo ejemplar de cada producto). Los productos tienen distintos precios entre 1 y k . Pero por ejemplo si se gasta 1\$ en el primer producto, y todos los demás cuestan $k\$$, no podrá comprar nada más.

1. Muestre que no se puede ser mejor que k -competitivo para este problema.
2. Considere ahora una variante en la que usted puede devolver uno o más productos ya comprados y recuperar el dinero (pero no puede comprar algo por lo que ya pasó). Muestre un algoritmo 2-competitivo para este caso.

P4 (1.0 pt)

Dado un grafo no dirigido $G = (V, E)$, el problema de MAX-CUT es el de dividir $V = S \cup S'$, con $S' = V - S$, de manera de maximizar la cantidad de aristas que cruzan de S a S' .

1. Considere el algoritmo que toma cada vértice y tira una moneda para decidir si lo pone en S o S' . Pruebe que, en el caso esperado, se obtiene una 2-aproximación.
2. Considere el algoritmo determinístico que toma cada vértice v_i , para $i = 1, \dots, |V|$, y lo pone en S o en S' según en cuál de los dos se maximice la cantidad de vértices entre v_i y v_1, \dots, v_{i-1} que cruzan de S a S' . Pruebe que este algoritmo es una 2-aproximación.

Tiempo: 3 horas