

CC4102 - Examen

Prof. Gonzalo Navarro

3 de Diciembre de 2014

P1 (2 pt)

Indique, en los siguientes casos, la única respuesta correcta a la pregunta “¿de qué está hablando esto?”. Los desaciertos no restan puntos.

1. Obtener la mediana determinísticamente requiere al menos $2,95n$ comparaciones.
 - a) Algoritmos aproximados
 - b) Algoritmos probabilísticos
 - c) Cotas inferiores
 - d) Algoritmos paralelos
 - e) Cotas superiores
2. Se puede obtener la mediana con $1,5n + o(n)$ comparaciones esperadas.
 - a) Cotas inferiores
 - b) Algoritmos en memoria secundaria
 - c) Algoritmos aproximados
 - d) Algoritmos aleatorizados
 - e) Análisis amortizado
3. Un índice que encuentra los k elementos más cercanos a una consulta se construye eligiendo K objetos del conjunto al azar y almacenando, para cada uno de ellos...
 - a) Algoritmos paralelos
 - b) Algoritmos aleatorizados
 - c) Algoritmos probabilísticos
 - d) Análisis amortizado
 - e) Algoritmos aproximados
4. El algoritmo para construir el árbol de sufijos de $T[1, n]$ procede de izquierda a derecha en T . Para cada nuevo carácter $T[i]$, extiende todos los nodos internos terminados en $T[i - 1]$. En total el costo es $O(n)$ porque...
 - a) Algoritmos aleatorizados
 - b) Cotas inferiores
 - c) Análisis amortizado
 - d) Algoritmos en memoria secundaria

- e) Algoritmos aproximados
5. Cuando se tiene el arreglo de n elementos, se busca cada uno de ellos en el B-tree, con lo cual el costo total es $O(n \log_B n)$.
- a) Análisis amortizado
 - b) Algoritmos aproximados
 - c) Algoritmos en memoria secundaria
 - d) Algoritmos en strings
 - e) Algoritmos paralelos
6. Con este método se logra eficiencia $O(1/\log n)$, la cual se puede mejorar reduciendo p .
- a) Cotas inferiores
 - b) Algoritmos aproximados
 - c) Algoritmos paralelos
 - d) Algoritmos aleatorizados
 - e) Algoritmos en strings

P2 (2 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. (1 pt) 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. (1 pt) 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.

P3 (2 pt)

Se tiene un árbol escrito en un arreglo $A[1..n]$, donde cada elemento $A[i] = j$ indica que el padre de i es j . La raíz r indica $A[r] = 0$.

1. (1 pt) Diseñe un algoritmo PRAM para calcular la profundidad de cada nodo (distancia a la raíz) en un arreglo $P[1..n]$, en tiempo $O(\log n)$. Analice $T(n)$, $W(n)$, speedup, eficiencia, y número óptimo de procesadores. ¿Qué tipo de modelo usó? (EREW, CREW, ...).
2. (1 pt) Lo mismo, ahora para calcular la altura de cada nodo (distancia a la hoja más profunda que descende de él). Considere un modelo CRCW donde, si hay varias escrituras concurrentes, la celda se queda con el máximo valor escrito.

Tiempo: 3 horas

Con tres hojas de apuntes