Auxiliar 12 - Problemas para el Control

CC4102 - Diseño y Análisis de Algoritmos
Profesor: Gonzalo Navarro Auxiliar: Jorge Bahamonde

3 de Diciembre del 2015

1 Análisis Amortizado: Árboles α-balanceados

Un *árbol α-balanceado*, para $1/2 < \alpha < 1$, es un árbol binario de búsqueda donde todo subárbol $T = (root, T_l, T_r)$, cumple $|T_l| \leq \alpha|T|$ y $|T_r| \leq \alpha|T|$. Las operaciones para buscar y mantener un árbol α-balanceado son las mismas que para un árbol binario de búsqueda, excepto que luego de insertar o borrar un nodo, se busca el nodo más alto en el camino del punto de inserción/borrado hacia la raíz, que no esté α-balanceado, y se lo reconstruye como árbol perfectamente balanceado (el costo es proporcional al tamaño del subárbol que se reconstruye).

1. Muestre que la búsqueda en un árbol α-balanceado cuesta $O(\log n)$, y que lo mismo ocurre con las inserciones y borrados, si no consideramos las reconstrucciones. ¿Qué constante obtiene multiplicando el $\log n$?

2. Muestre que el costo amortizado de las inserciones y borrados, ahora considerando las reconstrucciones, es también $O(\log n)$. Para ello, considere la función potencial

$$\Phi(T) = \frac{1}{2\alpha - 1} \sum_{T' \in T} \max\{||T'_l| - |T'_r|| - 1, 0\}$$

donde $T' \in T$ significa que T' es un subárbol de T. **Propuesto:** encontrar la constante que multiplica a este $\log n$, y recomendar un óptimo.

2 Dominios Discretos: Consultas de Rango

Describa un algoritmo que, dados n enteros en $[0, ..., k]$, preprocese su entrada y responda cuántos de estos enteros se encuentran en el rango $[a, ..., b]$ en tiempo constante. Su algoritmo debería tomar tiempo $\Theta(n + k)$ en el preprocesamiento.

3 Algoritmos Online: Tareas y Procesadores

Tenemos m procesadores idénticos. Como entrada recibimos una lista t_1, t_2,...,t_n de tareas con tiempos de procesamiento p_1, p_2,...,$p_n > 0$, respectivamente. Las tareas son recibidas secuencialmente, y sólo cuando una tarea llega conocemos su tiempo de procesamiento. Cada tarea debe ser asignada a una máquina inmediatamente, y la decisión no puede ser cambiada. La *carga* de un procesador es la suma de los tiempos de procesamiento de todas las tareas que le son asignadas. El costo de un algoritmo que resuelve el problema de asignacion es la máxima carga entre sus procesadores.

Considere el siguiente algoritmo para el problema anterior. Cada tarea es asignada al procesador con menor carga (en caso de empate, se elige cualquiera).

1. Demuestre que este algoritmo es $(2 - \frac{1}{m})$-competitivo.

2. Demuestre que esta cota es óptima para el algoritmo.

4 Algoritmos Probabilísticos: Asociatividad

Se nos entrega un conjunto X con n elementos y una operación binaria $\circ$. Se pide determinar si $\circ$ es asociativa en X: es decir, si para todos x, y, z en X, se cumple $(x \circ y) \circ z = x \circ (y \circ z)$. Diseñe un algoritmo (probabilístico) que resuelva este problema en $O(n^2)$ y con buena probabilidad.