

CC4102 - Diseño y Análisis de Algoritmos

Auxiliar 2

Prof. Jérémy Barbay; Aux. Mauricio Quezada

4 de Abril de 2012

1 R-Tree

Un *R-Tree* es un árbol, parecido al B-Tree, que almacena un conjunto de rectángulos en memoria secundaria. Se permite insertar y borrar rectángulos del conjunto. La consulta que se realiza es: dado un rectángulo C , encontrar todos aquellos que tienen intersección con C .

Cada nodo interno del R-Tree tiene varios hijos, y para cada hijo almacena el mínimo rectángulo que contiene a todos los rectángulos almacenados en las hojas que descienden de ese hijo (se llama MBR: minimum bounding rectangle). En las hojas se almacenan los rectángulos reales. Para resolver una consulta C , se ve en la raíz cuáles MBRs interseca C y se sigue recursivamente por esos hijos.

Con esta descripción sucinta y sin pedir más detalles:

1. Dé un pseudocódigo para el algoritmo de búsqueda. Explique por qué es mejor que los MBRs sean pequeños
2. Diseñe un algoritmo de inserción de rectángulos que requiera una cantidad logarítmica de I/Os. Discuta los pro y contra de las alternativas.
3. Diseñe un algoritmo de borrado de rectángulos que requiera una cantidad logarítmica de I/Os una vez que la hoja que lo contiene está identificada. Discuta los pro y contra de las alternativas.
4. *Propuesto 1.* ¿Cómo utilizaría un R-Tree para soportar consultas del tipo “saber la ubicación de todos los locales comerciales a 3km de mi ubicación actual”?
5. *Propuesto 2.* Suponiendo que en cada nodo existe una posibilidad p de que la consulta C interseque un rectángulo, demuestre que el costo promedio de una consulta es de la forma $O(n^\alpha)$, para un cierto $0 < \alpha < 1$.

2 Multiplicación de Matrices

Considere el problema de determinar, dada una secuencia de n matrices A_1, A_2, \dots, A_n , donde la matriz A_i es de dimensión $p_{i-1} \times p_i$, $i \in \{1, \dots, n\}$, el orden de multiplicación que minimiza el número de operaciones a realizar.

Diseñe un algoritmo que tome tiempo $O(n^3)$ que resuelva este problema. Note que el algoritmo no debe realizar las multiplicaciones, sino que entregar el orden adecuado.

3 van Emde Boas

1. Resuelva la recurrencia $T(u) = T(\sqrt{u}) + O(1)$
2. Queremos crear una estructura de datos con las siguientes propiedades:
 - Mantener un conjunto dinámico S de tamaño n de un universo de valores $U = \{0, 1, \dots, u - 1\}$ de tamaño u .
 - Soportar las operaciones:
 - $\text{Insert}(x \in U \notin S)$
 - $\text{Delete}(x \in S)$
 - $\text{Successor}(x \in U)$
 - $\text{Predecessor}(x \in U)$

Describa la estructura árbol de van Emde Boas y muestre que el tiempo de sus operaciones es $O(\lg \lg u)$.