Trabajo Práctico
Estructuras de Datos para RI

Bibliografía: [MIR] Capítulo 8, [SE] Capítulo 5, [MAN] Capítulo 4

1) Escriba un programa que tome un conjunto de documentos de un directorio, extraiga los términos y arme los índices que permitan soportar búsquedas mediante el modelo booleano. Utilice una lista de posteo sobre un archivo secuencial. (puede utilizar la librería Perl Tokenize). Luego, codifique un segundo programa que permita buscar por uno o dos términos utilizando los operadores AND, OR y NOT.

2) Utilizando el programa anterior ejecute corridas con diferentes colecciones. Calcule los tamaños mínimos, máximos y promedio de las listas de posteo. ¿Qué utilidad tiene esta información? Calcule la relación de overhead de los índices respecto de la colección. Calcule el overhead para cada documento. Luego, determine mínimos, máximos y promedio. ¿Qué conclusiones se pueden extraer?

3) Agregue documentos a una colección (indexación incremental) y repita el ejercicio 2. Sus resultados: ¿Son consistentes con la ley de Heaps? Este proceso es costoso: ¿Cómo se puede realizar eficientemente?

4) Modifique el programa del ejercicio 1 para armar un archivo invertido posicional a nivel de palabra. Luego, implemente consultas con operadores de proximidad.

5) Modifique el programa del ejercicio 1 para armar un archivo invertido con información de frecuencias. Luego, implemente consultas utilizando el modelo vectorial utilizando tres esquemas de ponderación y/o ranking diferentes.

6) Modifique su programa anterior para que realice indexación posicional y soporte búsquedas booleanas por frases.

7) Agregue skip lists a su índice del ejercicio 2 y ejecute un conjunto de consultas sobre el índice original y luego usando los punteros. Compare los tiempos de ejecución.

8) A partir de un conjunto de posting lists (en formato texto) provistas realice un programa que arme el vocabulario utilizando un B+Tree por un lado y un archivo binario con la información de las postings y frecuencias por el otro. Utilice el modelo de bloques, use DGaps y agregue skip-lists.

9) Sobre la estructura del ejercicio anterior escriba un programa que realice una evaluación Term-at-a-Time y otro usando Document-at-a-Time. Compare los tiempos de ejecución para un conjunto de queries dados. Separe su análisis por longitud de queries y de posting lists.