

Costo de operadores

Operación selección

En la altura del árbol n es la cantidad de registros que entran en un bloque, y K la cantidad de valores diferentes de la clave de búsqueda.

agoritmo	Condicion	costo (peor caso)	Aclaracion
Busqueda lineal		b_r	
Busqueda binaria	Si el campo de la seleccion es la calve con la que esta ordenado el archivo	$\log_2(b_r)$	
A3	Indice primario, igualdad basada en la clave	$\log(n/2)(K) + 1$	
A4	Indice primario, igualdad en atributo no clave	$\log(n/2)(K) + b_s$	b_s cantidad de bloques conteniendo registros con la clave de busqueda especificada que son leidos
A5	Indice secundario, igualdad en clave	$h_i + 1$	
A5bis	Indice secundario, igualdad en no clave	$h_i * r_s$	r_s cantidad de registros que cumplen la condicion.

Involucrando comparaciones

Selecciones de la forma $\sigma_{(A \leq V)}$ o $\sigma_{A \geq V}$

Se puede hacer con un escaneo lineal o usando indices.

Para los casos de indice secundario la búsqueda lineal puede ser mas barata. E implica una E/S para cada registro.

Condicion	costo(en bloques)	aclaracion
Indice primario con tabla ordenada en A para \geq	$h_i + b$	b = numero de bloques conteniendo registros $A \geq V$
Para \leq	se escanea la tabla secuencial-mente = b	b = cantidad de bloques de la tabla
Indice secundario para $A \geq V$	$h_i + 2n$	n cantidad de registros que cumplen la condición. Se usa la primera entrada de indice $\geq V$, se recorren los indices secuencialmente
Indice secundario para $A \leq V$	$h_i + 2n$	n cantidad de registros que cumplen la condicion. Se recorren los indices secuencialmente hasta la primera

Condicion	costo(en bloques)	aclaracion
		entrada > V.

Operación de proyección

- Requiere recorrer todos los registros y realizar una proyección de cada uno. -> Se recorren todos los bloques de la tabla. Costo:

$$b_r = \text{cantidad de bloques de tabla r}$$

Operación ordenación

La ordenación depende de a cantidad de bloques disponibles en la memoria principal. En el peor caso entra un bloque en memoria. Luego el coste es

$$b_r(2\lceil \log_{M-1}(b_r/M) \rceil + 1)$$

donde b_r cantidad de bloques de la tabla

donde M cantidad de bloques que entran en memoria

Operación reunión selectiva

Algoritmo	costo(Peor caso)	costo(mejor caso)
Reunion selectiva de loop anidado	$n_r * b_s + b_r$	$b_s + b_r$
Loop anidado por bloques	$\lceil b_r / (M - 2) \rceil * b_s + b_r$	-
Loop anidado indexado	-	-
Merge-sort	$b_r + b_s + \text{Costo de ordenar si no lo estan}$	-

Agregación

El costo es el mismo que eliminación de duplicados

Concatenación, intersección y resta

El costo para estos casos es el mismo

$$b_r + b_s$$