# Homework 10

Given,

T(R1) = 400, T(R2) = 500, T(R3) = 1000

V (R1, a) = 50, V (R1, b) = 50, V (R2, b) = 40

V (R2, c) = 100, V (R3, c) = 50, V (R3, d) = 100

## 1.1

T(Q) = T(R1)/V (R1, a) = 400/50 = 8

## 1.2

T(Q) = T(R1) \*(c2-c1)/D = 400 \* (50-10)/50 = 320

## 1.3

T(Q) = T(Q)/V (R1, b) = 320/50 = 6

## 1.4

T(Q) = (T(R1) \* T(R2))/max (V(R1, b), V(R2, b)) = (400 \* 500)/50 = 4000

## 1.5

T(Q) = (T(R1) \* T(R2) \* T(R3))/max (V(R2, c), V(R3, c))

= (400 \* 500 \* 1000)/(50 \* 100) = 40000

## 1.6

T(Q1) = T(Q) \* T(R2)/max(V(R1, b), V(R2, b)) = 320 \* 500 / 50 = 3200

T(Q2) = T(Q1) \* T(R3)/max (V(R2, c), V(R3, c)) = 3200 \* 1000/ 100 = 32000

## 2.1.a

The best query plan is to use the clustered index on E.title.

Cost = 2 (read root) + 10000 \* 10% (read the leaf level for CFO) + 2500 \* 10% (read the data pages for CFO)

Cost = 2 + 10000 ∗ 10% + 2500 ∗ 10% = 1252

## 2.1.b

An unclustered index would preclude the low cost of the previous plan and necessitate the choice of a simple filescan, cost = 10000, as the best.

## 2.1.c

Although the order of the B+ index key makes the tree much less useful, the leaves can still be scanned in an index-only scan, and the increased number of tuples per page lowers the I/O cost.

Cost = 10000 \* 5 = 5000.

## 2.2.a

Although this index does contain the output field, the dname still must be retrieved from the relational data pages, for a cost of 2 + 10000 ∗ 10% + 5000 ∗ 10% = 1502.

## 2.2.b

However, as the clustered B+ tree’s index contains all the indexes needed for the query and has a smaller tuple, scanning the leaves of the B+ tree is the best plan, costing 10000 ∗ .75 = 7500 I/O s.