Assignment 3

Yixuan Lu

301336734

Q1

a) Use primary dense B+ tree index on {ptype, pnumber}. Select candidate Key.
   cost = tree height + 1 = 5

b) Use primary dense B+ tree index on {ptype, pnumber}. Average number of records that have ptype = 'ABC' is 250000/500=500. So, average number of blocks needed is 500/10=50
   cost=tree height +50=54

c) Use extensible hash index on pid. Read directory resides on two disk blocks, read two buckets and read two records.
   cost=2+2+2=6

d) Use B+ tree index on {mcountry, mcity}. Average number of records that have mcountry = 'UK' and mcity = 'Sheffield' is 250000/2500=100. $100/30 \approx 4$ to read leaf node.
   cost=tree height+4-1+100=108

e) Scan B(Product)
   cost=25000

f) Use linear hash index on manufacturer. Average number of records for each manufacturer is 250000/ 10000= 25. Read 1 bucket and 25 keys.
   Use B+ tree index on pname. Average number of records that pname = 'foo345' is 250000/50000=5. Read 5 leaf nodes and 4 interior nodes.
   cost= (1+25) +5+4=35.

g) Use B+ tree index on {mcountry, mcity}. Need 250000/40=6250 records. Read 6250/30≈209 leaf nodes.
   cost=tree hight+209-1+6250=6463

h) Scan B(Product)
   cost=25000

i) Use primary dense B+ tree index on {ptype, pnumber}. Average number of records that have ptype = 'HJK' is 250000/500=500. So, average number of blocks needed is 500/10=50. Use extensible hash index on pid. Read directory resides on two disk blocks, read one bucket and read one record.
   cost=4+50 + 2+1+1=58

j) Use external merge sort.
   $2B(\text{ceiling}(\log_{M-1}\text{ceiling}(B/M)) +1) = 2*25000*3=150000$

k) Read all 25000 blocks. Ptype and decription are 96*25000 bytes. Since pages are 4096 bytes, we need to write (96+4) *250000/4096≈6104 pages.
   cost=25000+6104=31104

l) Read all 25000 blocks. Ptype and decription are 96*25000 bytes. We need to write (96+4) *250000/4096≈6104 pages with duplicate. We need ceiling (6104/50) ≈123 number of merge pass. Number of merges is ceiling($\log_{M-1}$(123)) ≈2. Total cost is 25000+ (2+1) *6104= 43,312

Q2

a) 180000
b) 180000/ (18000*2) =5 records per block
c) sqrt(B(V)+B(P)) ≈445 since main memory must be greater than sqrt(B(V)+B(P)) to perform the join in two passes. Main memory (in frames)>=445.
d) No change. We still need to compare records in V with records in P until all records have been read. So, P is sorted or not does not important for sort-join algorithm.
e) Partition smaller relation V to make sure that M-2>sqrt(B(V)) ≈135.
So, main memory (in frames) >=137.
f) Simple nested loop joins read V for each record. Use larger relation V in outer loop and smaller relation P in inner loop.
cost=B(P)+(ceiling(B(P)/(M-2) *B(V)) = 18000+ceiling (18000/1198) *180000≈2718000
g) Since the directory page of the index is retained in main memory, index cost for each record in relation P is 1+1 (bucket and record). According to Cost= B(V)+(T(V)*<index cost>) and visit is outer relation, cost=180000+1800000*2= 3,780,000.
h) One patient record match 10 visit record. Blocking factor is 10. And there is possible that one record resides on two blocks. Then there are two keys in one leaf node. So, index cost is tree height -1(directory page of the index is retained in main memory) + 10/10 +1 = 4
cost= B(P)+T(P)*<index cost>=18000+180000*4= 738,000
i) Partition Patient into k=ceiling (18000/4000) =5 partitions and Visit into 5 partitions. Only 1 partition can be retained in main memory. So, t=1. Cost is B(V)+B(P)+2*((k-t)/k) *(B(V)+B(T)) = 514,800