HW7
Name: Xiang Gu
EID: xg2847

Part A

15.3.2
10,000/999 * (999 + 10,000) \appro 10 * 11,000 = 110,000 disk I/O’s

15.3.3
a). 10,000/(M-1) * (M-1 + 10,000) <= 100,000 ⇔ M >= 1112

15.4.2
a). set union: 3 * (B(R) + B(S)) = 3 * 20,000 = 60,000 disk I/O’s

b). simple sort-join: 5 * (B(R) + B(S)) = 5 * 20,000 = 100,000 disk I/O’s

c). Merge-join: 3 * (B(R) + B(S)) = 3 * 20,000 = 60,000 disk I/O’s

15.4.3
We can modify phase 1 such that some blocks are not written back to secondary storage, which was written out and read back to main-memory in the original two-pass algorithm.
Concretely, we can
1). Set a smaller chuck size to read into main-memory in phase 1;
2). Perform the same sort process as before;
3). For the last chuck, do not write it back to secondary storage. Just leave it in main memory;
4). In phase 2, do the same thing as before but acknowledging the fact that what was originally the last sublist is now already in main-memory.

Part B

16.4.1
a). T(W join X join Y join Z) = 100 * 200 * 300 * 400 / (60 * 100 * 50) = 8,000

b). T(select_{c=20}(Y) join Z) = T(select_{c=20}(Y) * T(Z) / max{V(select_{c=20}(Y), d), V(Z,d)} = T(Y)/V(Y,c) * T(Z) / max{V(Y,d), V(Z,d)} = 300/50 * 400 / 50 = 48

c). T(X join{X.c<Y.c} Y) = T(select_{X.c<Y.c}(X CartisianProduct Y)) = T(X CartisianProdcut Y) / 3 = T(X) * T(Y) / 3 = 200 * 300 / 3 = 20,000

Part C

1.

a). Restricting to only left-deep trees:

Set	{W}	{X}	{Y}	{Z}		
Cost	0	0	0	0		
Best Plan	W	X	Y	Z		
Set	{W, X}	{W, Y}	{W, Z}	{X, Y}	{X, Z}	{Y, Z}
Cost	0	0	0	0	0	0
Best Plan	W join X	W join Y	W join Z	X join Y	X join Z	Y join Z
Set	{W, X, Y}	{W, X, Z}	{W, Y, Z}	{X, Y, Z}		
Cost	1000/3	1000/3	2400	600		
Best Plan	(W join X) join Y	(W join X) join Z	(Y join Z) join W	(X join Y) join Z		
Set	{W, X, Y, Z}					
Cost	1000 * 4/3					
Best Plan	((W join X) join Y) join Z					

b). Consider all trees:
(red cells are the only different cells in this answer compared to the above table)

Set	{W}	{X}	{Y}	{Z}		
Cost	0	0	0	0		
Best Plan	W	X	Y	Z		
Set	{W, X}	{W, Y}	{W, Z}	{X, Y}	{X, Z}	{Y, Z}
Cost	0	0	0	0	0	0
Best Plan	W join X	W join Y	W join Z	X join Y	X join Z	Y join Z
Set	{W, X, Y}	{W, X, Z}	{W, Y, Z}	{X, Y, Z}		
Cost	1000/3	1000/3	2400	600		
Best Plan	Y join (W join X)	(W join X) join Z	W join (Y join Z)	Z join (X join Y)		
Set	{W, X, Y, Z}					
Cost	1000 * 4/3					
Best Plan	Z join (Y join (W join X))					

2. In this problem, we restrict our attention to left-deep trees.
Query graph looks like (there is no query here but we will draw a link between two nodes if those two relations have shared attributes)

W------X------Y------Z

Copied from the table in Part C::1.a, red strikethrough cells are those eliminated by the query graph because the nodes are not adjacent, effectively reducing the size of the table from 15 entries to 10 entries.

Set	{W}	{X}	{Y}	{Z}		
Cost	0	0	0	0		
Best Plan	W	X	Y	Z		
Set	{W, X}	~~{W, Y}~~	~~{W, Z}~~	{X, Y}	~~{X, Z}~~	{Y, Z}
Cost	0	~~0~~	~~0~~	0	~~0~~	0
Best Plan	W join X	~~W join Y~~	~~W join Z~~	X join Y	~~X join Z~~	Y join Z
Set	{W, X, Y}	~~{W, X, Z}~~	~~{W, Y, Z}~~	{X, Y, Z}		
Cost	1000/3	~~1000/3~~	~~2400~~	600		
Best Plan	(W join X) join Y	~~(W join X) join Z~~	~~(Y join Z) join W~~	(X join Y) join Z		
Set	{W, X, Y, Z}					
Cost	1000 * 4/3					
Best Plan	((W join X) join Y) join Z					