The assignment is to be turned in before Midnight (by 11:59pm) on January 25th, 2018. You should turn in the solutions to this assignment as a pdf file through the TEACH website. The solutions should be produced using editing software programs, such as LaTeX or Word, otherwise they will not be graded. Trees can be drawn on paper and scanned.

1: File Structures (1 point)

1. Consider a file with a large number of *Customer(id, name, birth-date)* records. Assume that users frequently search this file based on a the field *id* to find the values of *name* or *birth-date* for customers whose information is stored in the file. Moreover, assume that users rarely update current records or insert new records to the file. Which file structure, heap versus sorted, provides the fastest total running time for users' queries over this file? Explain your answer (0.5 point).
2. Consider a file with a large number of *Transaction(id, customerID, productID, amount)* records, which keeps track of the purchases made by a customer on various products. Assume that users frequently insert new records into this file. Users also query this file to compute the total amount of money each customer has spent on her purchases, similar to a SQL query with *Group By customerID*. Which file structure, heap versus sorted, provides the fastest total running time for users' queries over this file? Explain your answer (0.5 point).

2: B+ Tree Indexing (2 points)

Consider the B+ tree index shown in Figure 2. Each intermediate node can hold up to five pointers and four key values. Each leaf can hold up to four pointers to data, and leaf nodes are doubly linked as usual, although these links are not shown in the figure. Answer the following questions.

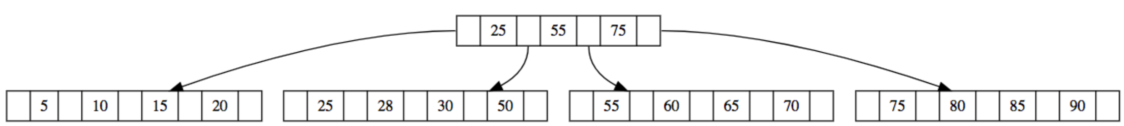

Figure 1: Tree for question 2.

1. Show the B+ tree that would result from inserting a record with search key 95 into the tree.
2. Use the result/solution tree from (1) and Show the B+ tree that would result from deleting the record with search key 60.
3. Name a search key value such that inserting it into the result/solution tree from (1) would cause an increase in the height of the tree.
4. What can you infer about the contents and the shape of A, B and C subtrees?

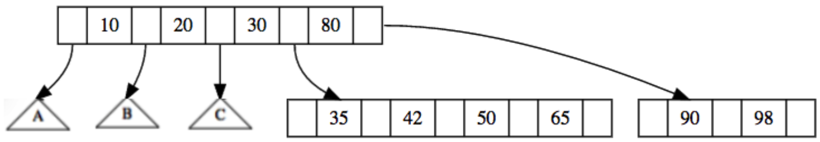

Figure 2: Tree for question 2.4.

3: B+ Tree Indexing (1 point)

Suppose that a block can contain at most four data values and that all data values are integers. Using only B+ trees of degree 2, give examples of each of the following:

1. A B+ tree whose height changes from 2 to 3 when the value 60 is inserted. Show your structure before and after the insertion.
2. A B+ tree in which the deletion of the value 60 leads to a redistribution. Show your structure before and after the deletion.

4: B+ Tree Indexing (1 point)

Consider the instance of the Students relation shown in Figure 3.

1. To reduce the number of I/O access in index search, each B+ tree node should fit in a block. Let *sid* be an integer requiring 16 bits. Let a pointer require 32 bits. If the block size is 28 bytes (consisting of 8 bits), what is the maximum degree of the B+ tree index on *sid* so each B+ tree node fit in a block?
2. Show a B+ tree index on *sid* of degree calculated in part 1 for all records in Figure 3.

sid	*name*	*login*	*age*	*gpa*
13822	James	james@music	11	1.8
13842	Lily	lily@music	12	3.8
13833	Jacob	jacob@music	12	3.8
13666	Jones	jones@toy	18	3.9
13667	Jones	jones@toy	18	3.9
13902	Jones	jones@toy	18	3.9
13904	Jones	jones@toy	18	3.9
13906	Jones	jones@toy	18	3.9
13908	Jones	jones@toy	18	3.9
13910	Jones	jones@toy	18	3.9
13652	Scott	scott@music	21	3.5
14001	Smith	smith@toy	19	3.4

Figure 3: An Instance of the Students Relation.