Homework3
CSC 675/CSC 775

Total points: 100

Q1. (8 points, 1 point each) Select either True or False for each of the following questions.

a) Each relation can have multiple clustered or unclustered indexes.
b) A relation can have a clustered index on a set of attributes.
c) Each relation can have multiple indexes using Alternative1.
d) Range queries cannot be answered using hash indexes.
e) Maintaining clustered indexes are more expensive than sorted files.
f) Index-olny plans work equally efficiently with unclustered indexes.
g) In any index file the data records are sorted by the search keys.
h) In any index file the data entries are sorted by the search keys.

Q2. (25 points, 5 points each) Consider the following instance of the Students relation, sorted by age. Assume that these tuples are stored in a sorted file in the order shown; the first tuple is on page 1, the second tuple is also on page 1; and so on. Each page can store up to three data records; so the fourth tuple is on page 2. Explain the data entries that each of the following indexes contain. If such an index cannot be constructed, provide reason.

Sid	Name	Login	Age	gpa
53831	Mary	May@music.com	11	2.8
53832	Sam	guldu@music.com	12	2.0
53660	George	Mary@cs.com	13	3.0
53666	Sara	jones@cs.com	18	3.4
53688	James	smith@ee.com	18	3.2

a) An unclustered index on age using Alternative (1).

b) A clustered index on <age, gpa> using Alternative (2).

c) An unclustered index on name using Alternative (2).

d) A clustered index on name using Alternative (2).

e) An unclustered index on gpa using Alternative (2)

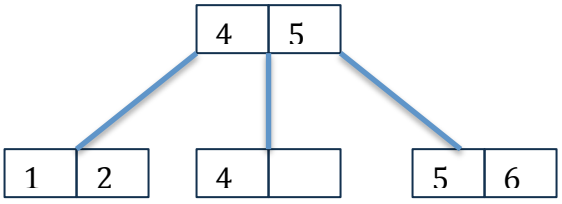
(25 points) Question 3. Given the following parameters:
1 Page on disk = 1024 bytes
Key = 26 bytes
Pointer = 4 bytes

a) (5 points) What is the best value for the entries m?

b) (5 points) How many I/O operations are required for searching through 1000,000 pages using an index file with m entries (m is calculated in part a)?

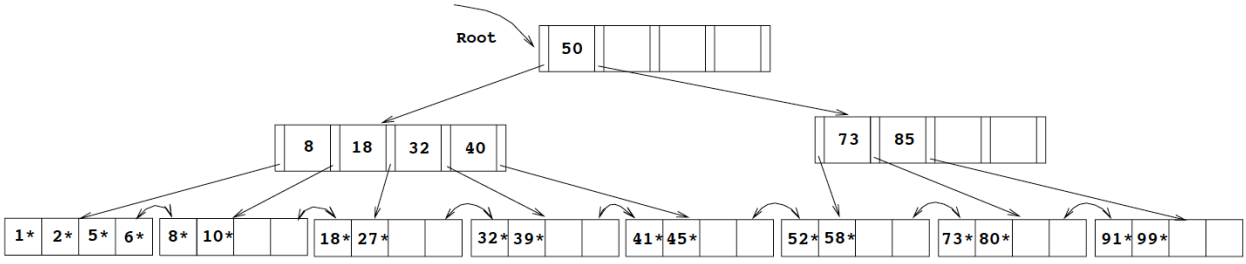
c) (5 points) How many I/O operations are required for searching through 1000,000 pages using a sorted file.

d) (10 points) Insert the value 3 into the following B+ tree with m=3.



(22 point) Question 4

Consider the B+ tree shown below. Assume the left order is for < and the right order is for >=. If you can borrow from both siblings, choose the right sibling. Answer these questions:



a) (10 points) Show the B+ tree after deleting the data entry with key 52 from the original tree.

b) (12 points) Show the B+ tree after successively deleting the data entries with keys 32, 39, 41, 45, and 73 from the original tree. Show only the final tree.

Question4 (20 points, 5 points each)
Consider the following relational schema as part of a university database:
Prof(sin, pname, office, age, sex, specialty, dept_did)
Dept(did, dname, budget, num_majors, chair_sin)

Assume the following queries are the most common queries in the workload for this university and they are almost equivalent in frequency and importance:
Assume that both B+ trees and hashed indexes are supported by the DBMS and that both single- and multiple-attribute index search keys are permitted.
Specify the attributes you recommend indexing on, indicating whether each index should be clustered or unclustered and whether it should be a B+ tree or a hashed index.
Note: If you can search with an index-only plan you should create a composite search key.

1. Find the maximum age of professors in each department.

2. Find the name of professors who have a user-specified research specialty.

3. List names of the departments with the budget more than 100K.

4. List the department id and chairperson for departments with a user-specified department name.