

CS304 Database System Concepts

Assignment 6

Due: March 27, 2012

(Please submit hard copies to class or to Zheng on due date.)

Name:

Matric No:

Q1. (2 points) Consider the following student file in figure 1, and assume $n=3$

<i>sid</i>	<i>name</i>	<i>login</i>	<i>age</i>	<i>gpa</i>
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	3.8
53666	Jones	jones@cs	18	3.4
53901	Jones	jones@toy	18	3.4
53902	Jones	jones@physics	18	3.4
53903	Jones	jones@english	18	3.4
53904	Jones	jones@genetics	18	3.4
53905	Jones	jones@astro	18	3.4
53906	Jones	jones@chem	18	3.4
53902	Jones	jones@sanitation	18	3.8
53688	Smith	smith@ee	19	3.2
53650	Smith	smith@math	19	3.8
54001	Smith	smith@ee	19	3.5
54005	Smith	smith@cs	19	3.8
54009	Smith	smith@astro	19	2.2

Figure 1

- Build a B^+ -tree index with search key *sid* for this file
- Build a B^+ -tree index with search key *gpa* for this file

Q2. (2 points) Suppose there is a relation $R(A,B,C)$, with a B^+ -tree index with search key (A,B) .

- a) What is the worst case cost of finding records satisfying $10 < A < 50$ using this index, in terms of the number of records retrieved n_1 and the height h of the tree?
- b) What is the worst case cost of finding records satisfying $10 < A < 50 \wedge 5 < B < 10$ using this index, in terms of the number of records n_2 that satisfy this selection, as well as n_1 and h defined above.
- c) Under what conditions on n_1 and n_2 would the index be an efficient way of finding records satisfying $10 < A < 50 \wedge 5 < B < 10$.

Q3. (2 points) Suppose that you have a sorted file and want to construct a B^+ -tree index on this file.

- a) One way to accomplish this task is to scan the file, record by record, inserting each entry using the B^+ -tree insertion procedure. What performance and storage utilization problems are there with this approach?
- b) Propose another method.

Q4. (2 points) Practice Exercise 12.7 on page 527 of Text book.

Q5. (2 points) Answer the following questions about Extendible Hashing

- a) After an insertion that causes the bucket address table size to double, how many buckets have exactly one bucket address table entry pointing to them? If an entry is then deleted from one of these buckets, what happens to the bucket address table size? Explain your answers briefly.
- b) Does Extendible Hashing guarantee at most one disk access to retrieve a record with a given key value?
- c) If the hash function distributes data entries over the space of bucket numbers in a very skewed (non-uniform) way, what can you say about the size of the bucket address table? What can you say about the space utilization in buckets?