Homework 2
Index Structures

Due: *Thursday, May 15, in-class*

Submission. I encourage everyone to use text-processing software for the solutions and use PowerPoint, xfig or other graphical tools to draw pictures/diagrams required in this homework. (I will accept handwritten solutions, but I discourage them. This policy is mainly to ensure expedient and error-free grading).

Problems

Consider a database relation R with an integer key X, which needs to be indexed. X has no duplicate keys. We consider the following two file organizations:

Data File A. Data file A is a sequential file with X as the search key. Each disk page can store three (3) records from the relation R on it.

Data File B. Data file B is a heap file. Each disk page can store two (2) records from the relation R on it.

Consider the following sequence of insertion and deletion operations for the relation R. For simplicity, `Insert(10)` means "insert a record with the value of X attribute set to 10".

```
Create(R);

Insert(10);
Insert(100);
Insert(75);
Insert(110);
Insert(80);

Insert(20);
Insert(35);
```

```
Delete(75);
Delete(20);
Insert(65);

Insert(150);
Insert(125);
Insert(15);
Delete(10);
Delete(35);


Insert(60);
Insert(55);
Delete(65);
Insert(165);
Delete(150);

Insert(75);
Insert(10);
Insert(40);
Delete(110);
Insert(95);
```

Problem 1

For each data file structure (A and B), show the state of the data files after each five `Insert`/`Delete` operations. For each disk page, show only pageID and any pointers you may have in the header (nothing else is needed for this exercise). You can omit the header page - just show the pages with the data.

Problem 2

For Data File A, show the state of the following index files after every five `Insert`/`Delete` operations:

1. Simple Dense index (6 index records per page) on X;
2. Simple Sparse (6 index records per page) index on X.
3. Dense B+-tree (3 key values per page) index on X;

Problem 3

For Data File B, show the state of the following index files after every five `InsertDelete` operations:

1. Simple secondary index (5 index records per page) on X;
2. B+-tree secondary index (4 key values per page) on X;