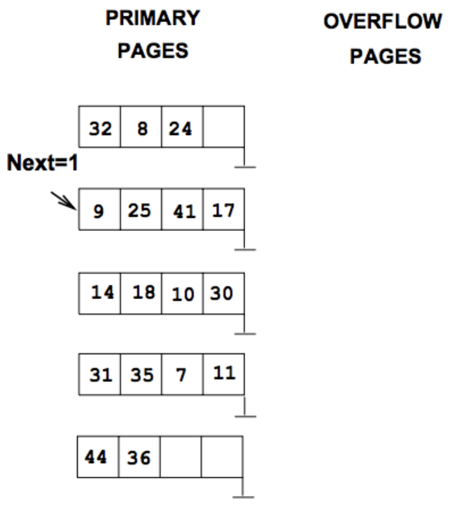
Boston University
CAS CS 562: Advanced Database Applications
Homework #1
Fall 2020

Due Date: Feb 20, 2020 at 11:59PM, please submit a PDF file using Gradescope

Problem 1. (Linear Hashing)



Consider the Linear Hashing index shown above. Assume that we split whenever an overflow page is created. Also, assume that we use $N = 4$ and the family functions of $h_i(x) = h(x) \ mod \ 2^i * N$. We initially start with $h_0 = h(x) \ mod \ 4$ and $h_1(x) = h(x) \ mod \ 8$. The capacity of each bucket (page) is 4.

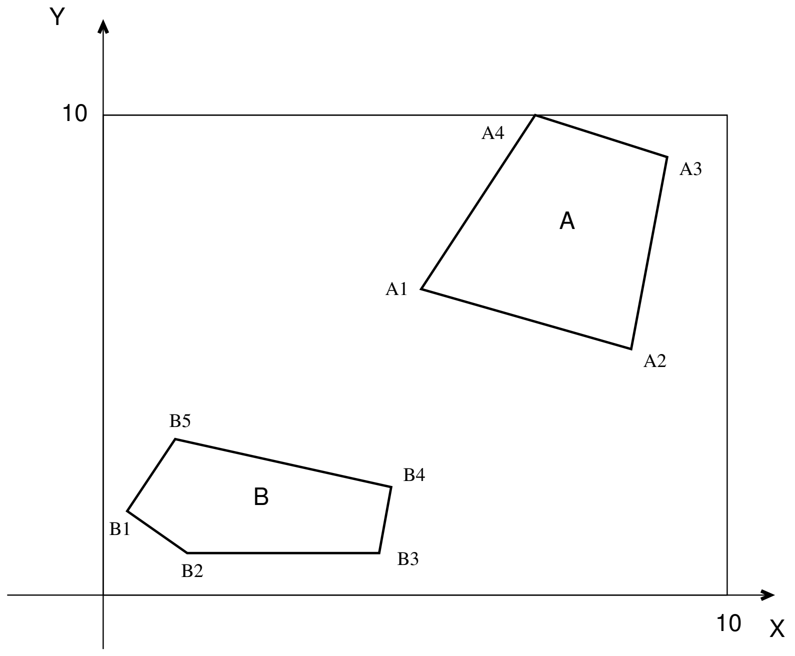
Answer the following questions about this index:

1. What can you say about the last entry that was inserted into the index?
2. What can you say about the last entry that was inserted into the index if you know that there have been no deletions from this index so far?
3. Suppose you know that there have been no deletions from this index so far. What can you say about the last entry whose insertion into the index caused a split?
4. Show the index after inserting an entry with hash value 4.
5. Show the index after inserting an entry with hash value 15 into the original index.
6. Find a list of entries whose insertion into the original index would lead to a bucket with two overflow pages. Use as few entries as possible to accomplish this. What is the maximum number of entries that can be inserted into this bucket before a split occurs that reduces the length of this overflow chain?

Problem 2. (Space Filling Curves)

a) Compute the Z-values and the Hilbert values for the points (10, 26) and (55, 63) for $K = 2$ and $K = 4$, where K is the number of bits per dimension.Assume that the data space is [0, 100) x [0, 100).

b) Compute the Z-values of regions **A** and **B** in the figure below, where A1=(5.5, 6.5), A2=(8.5, 5), A3=(9.1, 9.2), A4=(7, 10), B1=(0.4, 1.8), B2=(1.2, 0.6), B3=(4, 0.8), B4=(4.5, 2.3) and B5=(0.9, 3.2).Assume that $K = 4$ bits, the maximum number of bits per dimension ($2^4 = 16$ values per axis).



HINT: Methods to compute z- and Hilbert- values can be found in the following paper:
H. V. Jagadish: *Linear Clustering of Objects with Multiple Attributes.* **ACM SIGMOD Conference** 1990, pages 332-342.

Problem 3. (R-trees)

Let D be a 2-dimensional point dataset and $p = (x, y)$ a point in that set. The coordinates of all points are positive. Consider the function: $f(p) : D \to R$, where $f(p) = a_1x + a_2y$ and $a_1 + a_2 = 1$. The values for a_1 and a_2 are given by the user. The idea is that each user gives different importance (weight) to different attributes. We want to find the point (or points) that maximize(s) this function. This type of queries are called *preference* queries. Now, assume that an R-tree is used to store the dataset D.

(a) Design an efficient search procedure that uses the R-tree to find the point(s) that maximize the function f. Give the pseudo-code of the algorithm and explain how it works.

(b) What is the property that allows the design and guarantees the correctness of your algorithm? Explain.

Problem 4. (General)

Answer the following questions about some of the methods that we discussed so far:

(a) Consider a *range counting query*, where the query asks for the number of points inside a range. Assume that you have a dataset of 2-d points and you want to create an index based on kd-trees to answer efficiently range counting queries. Explain how you will modify the original kd-tree data structure and give an algorithm that will use this modified kd-tree to answer efficiently range counting queries. What is the worst case performance of your data structure for range counting queries?

(b) Discuss the different splitting policies of the Grid File and what are the pros and cons of each method. Which method you would use in your implementation? Why? Explain.

(c) What is the **paging algorithm** of the LSD-tree and how we use it?