15-415 Database Applications
Homework 6 Solutions

October 26, 2009

1 Question 1: B+ Trees

Q1.1 The height = 2.

Note: Many students mentioned height as 3. This is the number of *levels*, not the height. But we have accepted this answer.

Q1.2 Keys which cause a height increase = $\{2, 42\}$.

Common Mistake: Several students gave {2, 10, 42} as the answer. This is wrong as the key with value 2 never goes to the root. {2, 4, 6, 8} are directly inserted as a leaf (say L1). Meanwhile the root is empty except for the first child pointer to L1. The next key (10) then goes to the first slot in the root and so on. See the text for an illustration of the *bulk-loading* algorithm.

Q1.3 We need to merge three neighbors to produce two leaves. Figure 1 shows the resulting tree.

Q1.4 Any key in the range [53, 78] will cause a split that propagates all the way to the root.

2 Question 2: Extendible Hashing

Q2.1 Extendible hashes can be built taking the highest or lowest order bits. Both approaches are acceptable. Figure 2 shows the hash obtained after all the insertions.

Q2.2 Recall that we split whenever an insert happens on a bucket whose local depth is equal to the global depth. Hence, a maximum of 18 keys can be inserted without doubling for the specific hash.

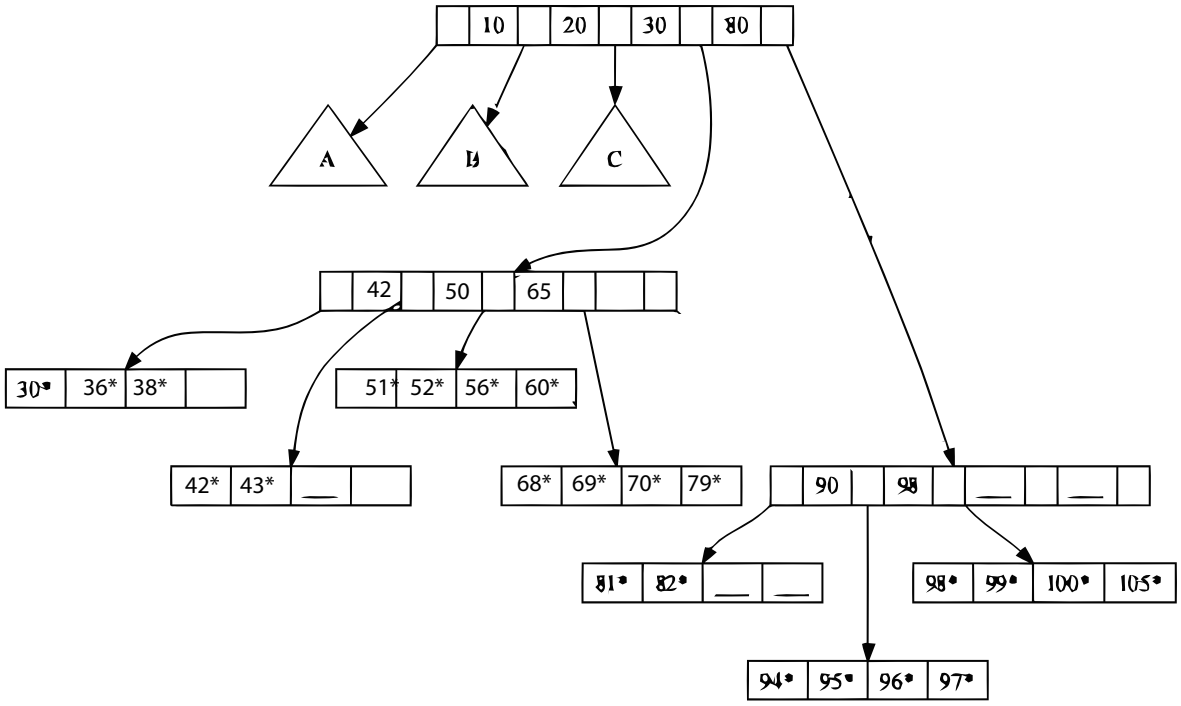

Figure 1: B+ tree for Q1.3

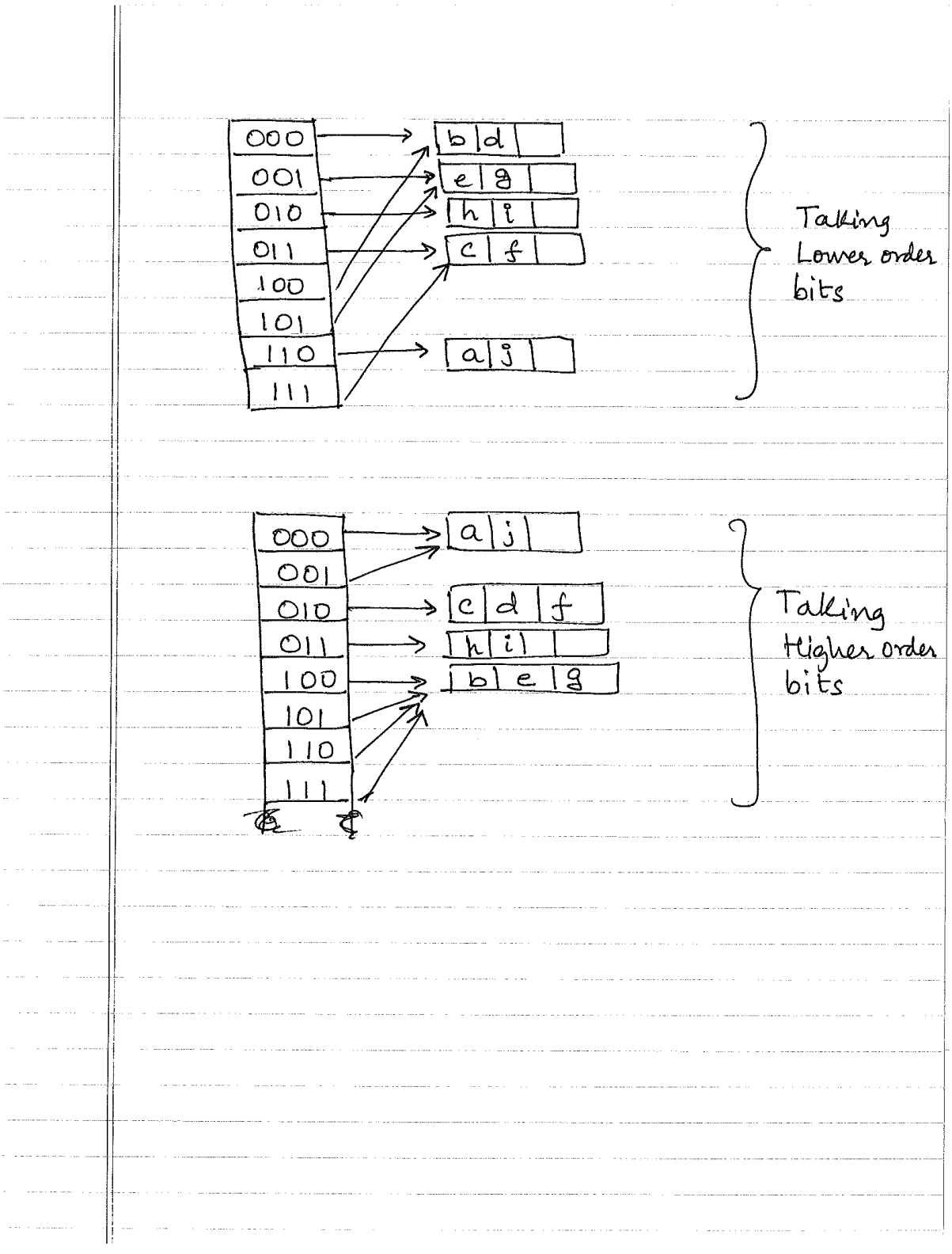

Figure 2: Extendible Hash for Q2.1

3 Question 3: Linear Hashing

Q3.1 Figure 3 shows the linear hash obtained after all the insertions.

Q3.2 A linear hash contracts only when the *last* bucket becomes empty. So, in this case, nothing changes much - just the appropriate keys are deleted, the `Next` pointer still remains at 0 and there would be NO overflow pages remaining (the key 50 would go into the primary pages after the deletion of key 66).

4 Question 4: External Sorting

The idea is that the total time taken is = Number of passes * (R + W) * Number of pages, where

- R = Time to read one page (affected by whether the input buffers have multiple pages as then we can do sequential I/O).
- W = Time to write one page (again, similarly affected by whether the output buffers have multiple pages or not).
- Number of passes = $\lceil \log_k(N1) \rceil + 1$ (as given in textbook) (depends only on *total* number of buffer pages, size of file and number of merges (k-way) we want to do).

Q4.1 The time to read/write each page = $10 + 5 + 1 = 16$ms.
The total number of passes = $1 + \lceil \log_{319}(10^7/320) \rceil = 3$.
The total time = $3 \times 10^7 \times 2 \times 16ms = 9.6 \times 10^8$.

Q4.2 The time to read 1 page = $10 + 5 + 1 = 16ms$.
The time to write 64 pages = $10 + 5 + 64 = 79ms$.
The total number of passes = $1 + \lceil \log_{256}(10^7/320) \rceil = 3$.
The total time = $3 \times 10^7 \times (16 + 79/64)ms = 5.1 \times 10^8$.

Q4.3 The time to read 16 pages = $10 + 5 + 16 = 31ms$.
The time to write 64 pages = $79ms$ as before.
The total number of passes = $1 + \lceil \log_{16}(10^7/320) \rceil = 5$.
The total time = $5 \times 10^7 \times (31/16 + 79/64)ms = 1.6 \times 10^8$.

5 Question 5: Relational Operators

M = pages in R = 10,000
N = pages in S = 200,000
B = buffer pages = 1002

Q5.1 a. Nested Loops Join Cost = $M + M \cdot N$ = 10,000 + 10,000 * 200,000 = 2,000,010,000.

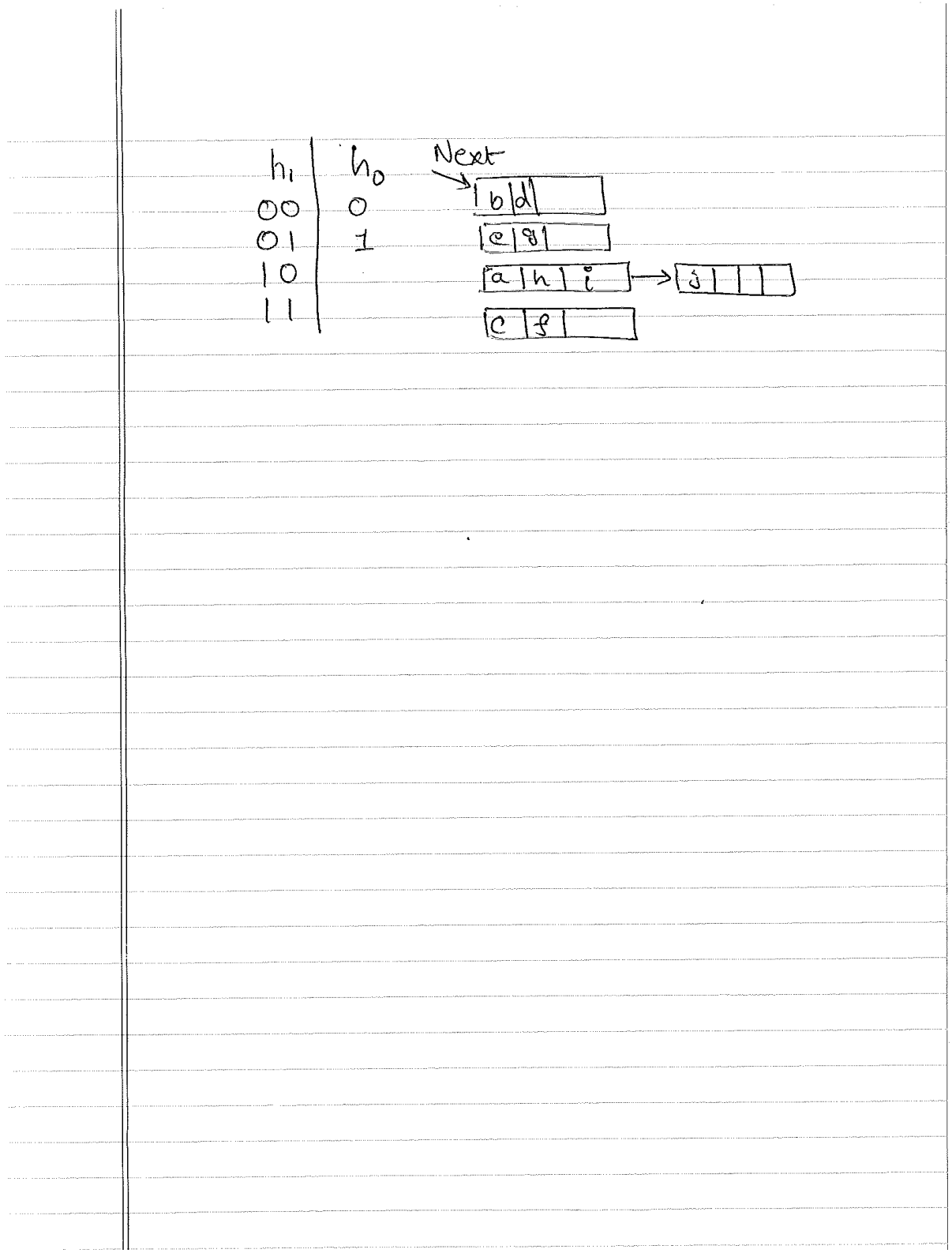

Figure 3: Linear Hash for Q3.1

b. Block-Nested Loops Join Cost = $M + N \cdot \lceil M/(B-2) \rceil$ = 2,010,000.

c. Sort-Merge Join Cost:
I/O cost of sorting R = $2 * 10,000 * (1 + \lceil \log_{1002-1}(10,000/1002) \rceil) = 40,000$
I/O cost of sorting R = $2 * 200,000 * (1 + \lceil \log_{1002-1}(200,000/1002) \rceil) = 800,000$
Total I/O cost = M + N + sorting R + sorting S = 10,000 + 200,000 + 40,000 + 800,000 = 1,050,000.

d. Hash Join Cost = $3 \cdot (M + N)$ = 630,000.

Q5.2 Yes, it is equivalent to finding the smallest B such that when $N = 200,000$, the number of passes, $1 + \lceil \log_{B-1}(N/B) \rceil = 2$. The answer is $B_{MIN}^{SMJ} = 448$. Interestingly, this $(\sqrt{N})$ is also the minimum buffer pages required so that the refinement of combining the merging phase of sorting with the merging phase of the join (as given in the book later in the section) can be used.

Q5.3 Yes. We need to have $B > \sqrt{(f \cdot M)}$, where f is the fudge factor. Generally f is very close to 1, so $\lceil \sqrt{M} \rceil + 1$ is a good estimate for B_{MIN}^{HJ}. The answer is 101.
Note: We have also accepted answers where students have just left it in terms of f.