CMSC 420: Fall 2022

Homework 4: B-Trees, Tries, and Memory Management

Handed out Tue, Dec 6. Due **Tue, Dec 13, noon**. Solutions will be released Tue, Dec 13 at noon, so no late submissions. Note that your lowest homework score will be dropped and the other three scaled up to 200 total points. So, this homework is effectively optional.

Problem 1. (18 points, 6 points each) Consider the B-trees of order 4 shown in Fig. 1 below. Recall that each non-leaf node has between 2 and 4 children, and every node has between 1 and 3 keys. Let us assume two conventions. First, key rotation (when possible) has precedence over splitting/merging. Second, when splitting a node, if the number of keys shared by the two new nodes is an odd number, the rightmost node receives the larger number of keys. **Note: An earlier version said "leftmost" instead of "rightmost", but this is not consistent with the lecture notes, so I changed it. We will grant full credit either way you do it, as long as you apply your rule uniformly.**

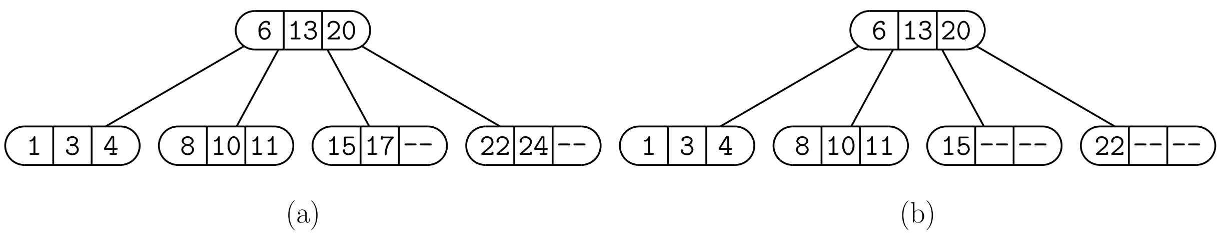

Figure 1: B-tree operations.

(a) Show the B-tree that results after inserting the key 9 into the tree of Fig. 1(a).

(b) Show the B-tree that results after inserting the key 2 into the (original) tree of Fig. 1(a).

(c) Show the B-tree that results after deleting the key 22 from the tree of Fig. 1(b).

(Intermediate results are not required, but may be given to help assigning partial credit.)

Problem 2. (16 points, 8 points each) In this problem we will build a suffix tree for $S =$ `"aababaabaab$"`.

(a) Recall that the 12 suffixes of S are (in reverse order):

$$S_{11} = \texttt{"\$"}, \quad S_{10} = \texttt{"b\$"}, \quad S_9 = \texttt{"ab\$"}, \quad \ldots, \quad S_0 = \texttt{"aababaabaab\$"}.$$

Let id_j denote the *substring identifier* for S_j. (Recall from Lecture 18 that this is defined to be the shortest prefix of S_j that uniquely identifies it.) List all 12 substring identifiers for these suffixes in index order (from first to last $\text{id}_0 \ldots \text{id}_{11}$).

(b) Draw the suffix tree for S. Draw your tree in the same edge labeling style we used in Fig. 7 in Lecture 18 LaTeX lecture notes. Order the children of each node in alphabetical order from left to right. (The form of your drawing is important. There are many online suffix-tree generators, and if it appears that you copied your answer from one of these, you will receive no credit.)

Hint: Begin by writing out all the substring identifiers in alphabetical order, one above the other. This makes it easy to determine common substrings.

Problem 3. (16 points, 8 points each) This problem involves performing operations using the buddy system for memory allocation.

(a) Consider the buddy allocation shown in Fig. 2. Explain which blocks are split in order to perform the operation `alloc(2)`. Show the final blocks and indicate what level of the structure they reside. Assume that we always split the leftmost block of sufficient size. You may assume that the size of the final block is exactly 2, there is no need to round the size up for the sake of adding header information. (You don't need to redraw everything, just the portion that changes.)

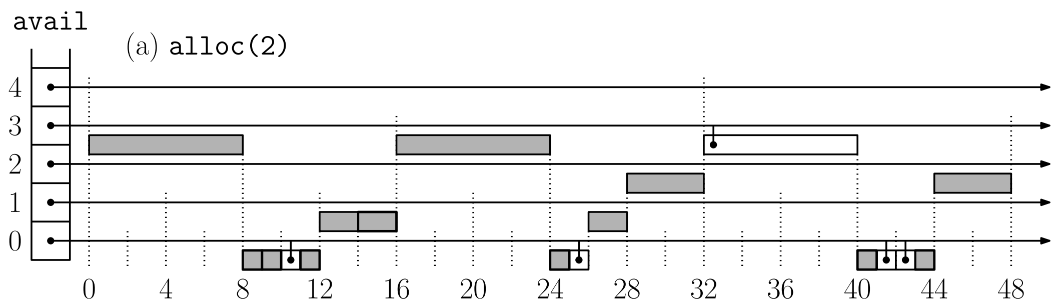

Figure 2: Problem 4(a): Buddy system allocation.

(b) Consider the buddy allocation shown in Fig. 3. Explain which blocks are merged in order to perform the operation `dealloc(24)`, which deallocates the shaded block of size 1 at address 24 as shown in the figure. Show the final merged block and indicate which level it resides at. (You don't need to redraw everything, just the portion that changes.)

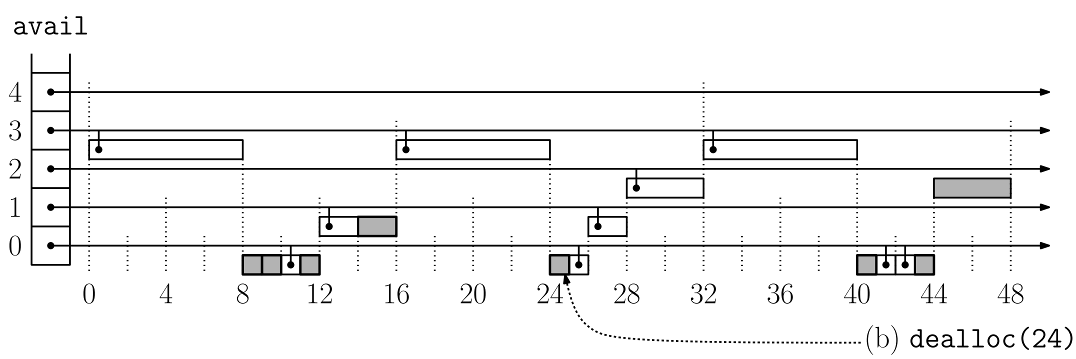

Figure 3: Problem 4(b): Buddy system deallocation.