# Dalhousie University
# CSCI 6057/4117 — Advanced Data Structures
# Winter 2022
# Assignment 3

*Distributed Wednesday, February 9 2022.*
*Due at 23:59 Wednesday, March 2 2022.*

**Guidelines:**

1. All assignments must be done individually. The solutions that you hand in must be your own work.
2. Submit a PDF file with your assignment solutions via Brightspace. We encourage you to typeset your solutions using LaTeX. However, you are free to use other software or submit scanned handwritten assignments as long as **they are legible**.
3. Working on the assignments of this course is a learning process and some questions are expected to be challenging. Start working on assignments early.
4. Whenever you are asked to prove something, give sufficient details in your proof. When it is straightforward to prove a lemma/property/observation, simply say so, but do not claim something to be straightforward when it is not.
5. We have the following late policy for course work: `http://web.cs.dal.ca/~mhe/csci6057/assignments.htm`

**Questions:**

1. [15 marks] Draw the x-fast trie that maintains the following set $S = \{0, 2, 8, 11, 14\}$ in universe $\{0, 1, 2, \ldots, 15\}$. Thus, $n = 5$ and $u = 16$.

   Your solution should include the tree structure, the descendant links, values stored in the leaves, and pointers between leaves.

   To show the dynamic perfect hash table constructed, simply give all the keys stored in the hash table.

2. [15 marks] This question asks you to perform competitive analysis of transpose (TR).

(i) [9 marks] Suppose that you are maintaining a list of $n$ elements under `access` operation only. The cost of `access` to the $i$-th element in the list is $i$. Let $S$ be a request sequence of $m$ `access` operations over this list.

For any sufficiently large $m$, construct a request sequence $S$ such that for this request sequence, the total cost of TR divided by the total cost of $S_{opt}$ is $\omega(1)$.

(ii) [6 marks] Use the result of (i) to argue that TR is not competitive.

3. [10 marks] In class, we discussed the problem of computing a static optimal binary search tree. Now we draw each binary search trees in a slightly different way by adding nodes representing failures in searches: We first draw the same tree structure. Then we label a node $i$ if it stores $A_i$. Next, we augment the tree by adding children to each node that has less than two children, so that each node in the original tree has two children in the augmented tree. Nodes added in this way are called *dummy nodes*. When performing the search for a value not in $\{A_1, A_2, \ldots, A_n\}$ using the augmented binary search tree, we will reach a dummy node, and thus each dummy node represents a range between two consecutive given values. We label a dummy node with $i$ if it represents the range $(A_i, A_{i+1})$. Thus $q_i$ is the probability of reaching a dummy node labeled $i$.

   Figure 1 re-draws the example shown in class, in which squares represent dummy nodes.

   Now prove the following statement: If $p_n = q_n = 0$, then an optimal binary search tree storing $A_1, A_2, \ldots, A_n$ with probabilities $p_1, \ldots, p_n, q_0, \ldots, q_n$ can be obtained by making the following change to any optimal binary search tree for $A_1, A_2, \ldots, A_{n-1}$ with probabilities $p_1, \ldots, p_{n-1}, q_0, \ldots, q_{n-1}$: simply replace the dummy node labeled $n-1$ with the structure shown in Figure 2.

4. [10 marks] Let $T$ be an arbitrary splay tree storing $n$ elements $A_1, A_2, \ldots, A_n$, where $A_1 \leq A_2 \leq \ldots \leq A_n$. We perform $n$ search operations in $T$, and the $i$th search operation looks for element $A_i$. That is, we search for items $A_1, A_2, \ldots, A_n$ one by one.

   (i) [5 marks] What will $T$ look like after all these $n$ operations are performed? For example, what will the shape of the tree be like? Which node stores $A_1$, which node stores $A_2$, etc.?

   (ii) [5 marks] Prove the answer you gave for (i) formally. Your proof should work no matter what the shape of $T$ was like before these operations.

   Hint: It may help to start with some specific examples and try to make some observations to make a guess. Then, construct a proof by induction.

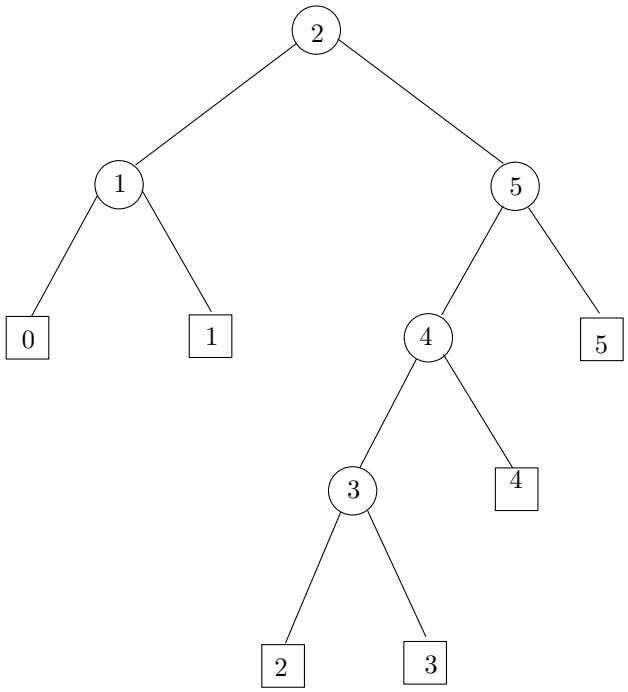


Figure 1: Re-drawing the example shown in class.

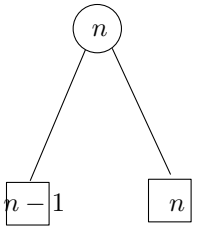


Figure 2: The structure used to replace the dummy node labeled $n-1$.