University of Waterloo
CS240E, Spring 2025
Assignment 4

Due Date: ~~**Tuesday, July 8, 2025**~~ **Thursday, July 10, 2025, 5pm**, with a grace period until 7:59pm

1 [20 marks]

Suppose that n is so large that we need external memory to store the n items of a priority queue. As usual, we use B to denote the size of a block of memory.

1. If you used binary heaps to realize ADT Priority Queue, what would be the number of block transfers for *insert*? Give a tight O-bound. No justification needed.
2. Recall from A1 that a d-ary heap (for integer $d \geq 2$) is a tree where every node has exactly d subtrees, all levels are full (except the last one, which is filled from the left), and the heap-order property holds.

   What is the height of a d-ary heap? Give a tight O-bound (depending on n and d), simplify as much as possible, and briefly **justify your answer**. You need not argue that the bound is tight. Note that d is not necessarily a constant.
3. Suppose that you want to use a d-ary heap to realize ADT Priority Queue in the external memory model such that operations use $O(\log_B n)$ block transfers. Explain how you would choose d (relative to B), and what gets stored in each block. Then explain how *delete-max* operates and why it uses $O(\log_B n)$ block transfers. (No need to analyze *insert*.)

2 [20 marks]

You are given a list S that stores n KVPs in sorted order. Describe an algorithm that has worst-case run-time $O(n)$ and that builds a skip-list L that stores the same KVPs. Furthermore, L should be *balanced* in the sense that searching in L has $O(\log n)$ *worst-case* time.

3 [20 marks]

Assume you are given a list L of n integers, sorted in increasing order. Show how to build a 2-4-tree that stores the items of L in $O(n)$ time.

4 [20 marks]

The worst case of interpolation search results in $\Theta(n)$ search time. Let i be the interpolated index value as predicted by the formula where the next comparison in the interpolation search is made. This splits the array into two parts at least one of which is of length no greater than half of the original range. We call this side the "good side" while the second part of length at least half the original range is called the "bad side". Assume we keep track of how many times after comparing with the element $A[i]$ interpolation search recurses on the bad side. Let this be a counter called numFailures. A recursion on the "bad side" is called an *interpolation failure*.

We wish to devise a new formula for the computation of the interpolated index that pushes the value i in such a way as to make the two parts more balanced, based upon the number of failures.

1. [3 marks] Give pseudocode for a modified interpolation search in which the two parts are made more balanced by reducing the size of the larger side by up to the value of numFailures. Make sure to handle boundary cases.

2. [5 marks] Give the worst case search time for this procedure.

3. [3 marks] Give pseudocode for a modified interpolation search in which the two parts are made more balanced by reducing the size of the larger side by up to the value of $2^{numFailures}$. Make sure to handle the boundary cases.

4. [5 marks] Give the worst case search time for this procedure.

5 [20 marks]

1. Suppose you implement skip lists with the probability that growing a tower is $1/t$ and the probability of stopping is $(t-1)/t$. If $t = 3$, then the probability of growing a tower by 1 is $\frac{1}{3}$; the probability of stopping $\frac{2}{3}$. So, for $t = 3$, what is the expected height, $E(h)$, of a tower such that each tower has height of at least one?

   Hint: In the problem, you will need to determine the value of S where $S = \sum_{h=1}^{\infty} \frac{h}{3^h}$. To do this, the mathematical "trick" is to consider the difference $3S - S$ and consider that one of the summations which appears in the difference is a geometric series. Recall that for any geometric series:

   $$\sum_{i=1}^{\infty} ar^{i-1} = \frac{a}{1-r}$$

   provided that $|r| < 1$. You may also find it helpful to know that the infinite series S is convergent and a closed-form solution for it exists.

2. The expected cost of a search is better for $t = 3$ than for $t = 2$ but worse for $t = 5$ than for $t = 2$. (In fact it is best for $t = e$.) Considering that the expected height of the tower continues to decrease as t increases, explain why the cost of search does not go down as t increases. (We are not expecting a mathematical analysis here, just a brief qualitative explanation.)

   Hint: If your written answer exceeds a half page of writing, one mark will be **deducted**. Please be concise. This question does not require a lengthy answer.