University of Waterloo
CS240E, Spring 2026
Midterm Post-Mortem

Question 1 [10 marks]

Correct answers earn 1 mark, incorrect answers earn -0.5 marks.

f) Midterm Review Q8

j) Increased by at most 2, not decreased.

Question 2 [8 marks]

d) Re-count the elements in the array. Note that the rank starts from 0, and there are **two** 30s in the array.

f) The bottommost layer is the original array - it does not need a coin flip. Also, a T is needed to terminate the coin flips.

g) The most common error is comparing x_3, which does not exist.

h) **Space** usage of a skip list is covered in lecture.

Question 3 [4+3+2+3+0+3+1+2+3+2+3+2+3 = 31 marks]

a) Most common mistake is second row, second column (False). Note: with generating functions (MATH239), you can resolve the recurrence here to an exponential.

b) Some of you have two flagged trees with the same height, or worked with $n = 2^k - 1$ instead of 2^k

c) Most submissions get it correct. Some lose points for answering the general purpose of randomization without giving an example

d) Some had incorrect answers - $O(n)$ was most common, which some justified by making the bound too loose (bounding both n_L and n_R by $n - 1$). Some had $O(1)$. Some of you got stuck at the recurrence - similar to expected length of downward walk in a binary tree, you will have to use Jensen's inequality because log(x) is concave.

e) Removed from midterm.

f) Most of you realized that it is $O(\log(n!))$, but did not simplify it to $O(n \log n)$.

g) For part (ii), note that each of the **leaves** are equally likely, we are taking the average over all leaves, not a random, relaxing walk down the decision tree.

h) There are two main approaches:

- Look at the rotations - single rotation, double rotation, work out that you have to have rotated at z's grandparent, and if you did a double rotation (which might cause grandchildren even if z is leaf) then the subtrees are empty. Partial marks are awarded based on progress. **(A lot of you missed the last step, and some had incomplete solutions)**
- (Cleaner) z's height decreased by at most 2, from h to $h-2$. The subtree rooted at where the rotation occured has height decreased from h to $h-1$. So z can have at most children.

i) There were many results which were not a treap. Most common errors include having duplicate priority of 2, and not being a BST (because the fix-up was not done via rotations).

j) Check for small computation mistakes (did you forget to -1 from the root node?)

k) If you think the statement is true, try $n = 3$ and see. This question was hinted at in lecture.

l) Some of you forgot that $\sum_{i=1}^{n} \frac{i}{2^i} \leq 2$ is in $\Theta(1)$.

Question 4 [(2+1)+5+3 = 11 marks]

a) There is an error in grading this question. It will be re-marked. (Mentioned in class)

b) In the explanation, a lot of you did not explicitly say that the subtree size will half at each step.

c) Many students gave an example without explanation.

Question 5 [2+3 = 5 marks]

a) You are asked to state "idea and analysis". Some of you did not analyze the runtime.

b) A lot of you did not do Q5b. For Q5b's key idea, go take a look at midterm review session Q15 (and maybe its solution).

Question 6 [6 marks]

Generally, if the data structure is unique given the input, or it depends on randomization that is independent of the input, it is oblivious.

b) Without rebalancing, the root is the first inserted node.

e) The last inserted node is splayed to the root.

f) Q3(h)

Question 7 [2+4+4+1+5+5 = 21 marks]

a) All fine.

b) Some solutions didn't mention how to bound the range we needed to look up. Some mistakenly used N as the range.

c)

- Most of you correctly got the idea to insert into B[0], and only merge when necessary / based on carries in the binary representation of n / similar to a Binomial Heap.
- Suppose you merge to B[0] B[k-1] to B[k]. Using a heap / priority queue here to merge all k relevant arrays is too slow - it gives a runtime of $O(1 + s \log k)$ or $O(1 + s \log \log s)$.
- Some of you are in a time crunch, and made small mistakes in analysis, or did not have time to put the whole sum together. It may be easier to bound $merge(A1, A2) \in O(max(A1, A2))$ by $O(A1)$ because the two arrays are the same size, instead of dealing with an uglier sum.

d) Reminder that for "briefly verify", you still need to (briefly) show your work. One sentence or even any bit of evidence should suffice.

e) For those who made it and attempted this question:

- Some did not specify time units.
- A common mistake is forgetting to include the newly inserted item's potential in the final potential calculation. This may be why you ended up with O(1) amortized time complexity. Also, try to be more careful in computation, since it could be the cause of some calculation mistakes.

f) No common errors (the vast majority left it blank).

Question 8 [1 mark]

- a surprising number of students left it blank
- graders were fairly generous with accepting the answers (e.g., even brief descriptions of the optional content from tutorials received full credit, even if something actually does appear in some form but the student did not recognize it)
- lazy deletion, skip list analysis, splay-tree operations, computing asymptotic notation from the definition/limit-rule are very common answers