University of Waterloo
CS240E, Winter 2021
Assignment 3

Due Date: Wednesday, March 3, 2021 at 5pm

The integrity of the grade you receive in this course is very important to you and the University of Waterloo. As part of every assessment in this course you must read and sign an Academic Integrity Declaration (AID) before you start working on the assessment and submit it **before the deadline of March 3rd** along with your answers to the assignment; i.e. **read, sign and submit A03-AID.txt now or as soon as possible**. The agreement will indicate what you must do to ensure the integrity of your grade. If you are having difficulties with the assignment, course staff are there to help (provided it isn't last minute).

The Academic Integrity Declaration must be signed and submitted on time or the assessment will not be marked.

Please read `http://www.student.cs.uwaterloo.ca/~cs240e/w21/guidelines/guidelines.pdf` for guidelines on submission. **Each written question solution must be submitted individually to MarkUs as a PDF** with the corresponding file names: a3q1.pdf, a3q2.pdf, ... , a3q7.pdf .

It is a good idea to submit questions as you go so you aren't trying to create several PDF files at the last minute. **Remember, late assignments will not be marked but can be submitted to MarkUs after the deadline for feedback if you email cs240e@uwaterloo.ca and let the ISAs know to look for it. (#3).**

1. (3+3=6 marks)

   Suppose you have a skip list with n keys and only three levels:

   - List S_0 contains $-\infty < a_0 < \cdots < a_{n-1} < \infty$.
   - List S_1 has the sentinels and k keys, where k is an integer that divides n (so that $n = km$ for some integer m). We assume that (with the exceptions of $\pm\infty$) entries are evenly spread out, so S_1 contains $-\infty, a_0, a_m, a_{2m}, \ldots, a_{(k-1)m}, +\infty$.
   - List S_2 contains only sentinels.

   (a) What is the worst-case time for a query? Give a $\Theta(\,)$ expression involving k and n.

   (b) Given n, how should you choose k to minimize this worst case, and what does the worst case become? Give a $\Theta(\,)$ expression in terms of n.

2. (3 marks)

Let A be an unordered array with n distinct items $k_0, \dots, k_{n-1}$. Give an asymptotically tight Θ-bound on the expected access-cost if you put A in the optimal static order for the following probability distribution:

$$p_i = \frac{1}{(i+1)H_n} \text{ for } 0 \le i \le n-1 \text{ where } H_n = \sum_{j=1}^{n} \frac{1}{j}.$$

For example, for $n = 4$ we have $H_4 = \frac{25}{12}$ and the items would have probabilities $\langle \frac{12}{25}, \frac{6}{25}, \frac{4}{25}, \frac{3}{25} \rangle$.

3. (9(+5)+4=13(+5) marks)

This assignment asks you to compare the performance of the MTF-heuristic for binary search trees with splay trees.

(a) Let T be a binary search tree with n nodes and height $h = n-1$, i.e., T is a path from the root to a unique leaf x. Show that if we perform *splayTree::search*(k) for the key k at x, then the resulting tree T' has height at most $h/2 + c$ for some constant c. Make c as small as possible..

Hint: Show a bound on the height of the subtree rooted at x after you have done i operations.

(b) (Bonus) Create an example of a binary search tree T with n nodes and a sequence of $\Theta(n)$ operations *BST-MTF::search* for keys in T such that the total number of rotations is in $\Theta(n^2)$.

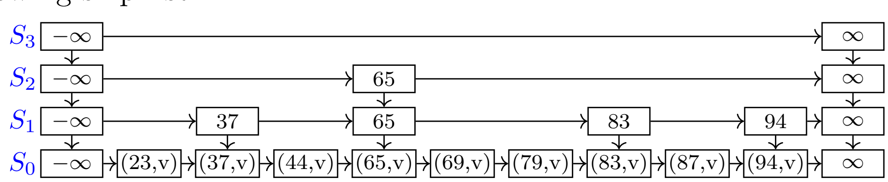
Hint: We had an example (cf. Module 05e or Figure 5.11 in the textbook-version from Feb. 2) where BST-MTF does badly. Start with that one, and continue to find bad search-keys. Can you describe the structure of the tree after each search?

(c) Prof. I.N.Correct claims that for any n they have an example of a binary search tree T with n nodes and a sequence of n operations *SplayTree::search* for keys in T such that the total number of rotations is in $\Theta(n^2)$. In particular the actual run-time for these n operations is in $\Omega(n^2)$.

Prove that this is impossible.

4. (5+3=8 marks)

In a skip list, let X be the total number of nodes that contain keys (i.e., $X = \sum_{i \ge 0} |S_i| = \sum_k X_k$). Correction: $X = \sum_{i \ge 0} |S_i| = \sum_k (1 + X_k)$. For example, $X = 14$ in the following skip list.



Note that X measures the space required for the skip list (when ignoring the sentinels). We showed in class that X has expected value $2n$. This assignment will guide you that with high probability, the actual space is close to this expected value.

(a) What is the variance of X? You should give a closed-form exact bound, but partial credit may be given for an upper bound.

Hint: Rather than computing the variance directly, you should apply some rules about variances and what they are for some known probability distributions. See Appendix A.

(b) Argue that $P(X \geq 4n) \in O(\frac{1}{n})$.

5. (6 marks)

Recall *interpolation-search* (Algorithm 6.2 from the textbook) and consider its performance for the sorted array $A[0..n-1]$ where $A[i] = ai + b$ for $0 \leq i \leq n-1$ (for some constants $a > 0$, and $b \in \mathbb{R}$). Show that then a search for a key k always takes $O(1)$ time, regardless of whether key k is in A or not.

6. (6 marks)

This question concerns sorting of infinite-precision numbers $x_0, \ldots, x_{n-1}$. Specifically, each x_i is in $[0, 1)$ and written in base-2. It is given to you implicitly, via an accessor-function *get-decimal-place*(i, d), which returns the bit in the dth decimal place of x_i. For example, if $x_i = 0.001001...$ then *get-decimal-place*$(i, 3) = 1$ and *get-decimal-place*$(i, 4) = 0$. Function *get-decimal-place* takes $\Theta(1)$ time.

Describe an algorithm to sort these (implicitly given) numbers $x_0, \ldots, x_{n-1}$ in $O(n \log n)$ expected time, assuming the numbers $x_0, \ldots, x_{n-1}$ have been randomly and uniformly chosen from the interval $[0, 1)$. You may also assume that all numbers are distinct. Note that comparing x_i and x_j is *not* a constant-time operation! Your output should be the sorting-permutation π (i.e., $x_{\pi(0)} < x_{\pi(1)} < \cdots < x_{\pi(n-1)}$).

A high-level description is enough, no need for pseudo-code, and the correctness can be extremely short. (But do argue the run-time carefully.)

Hint: There is a reason why this question is here and not on Assignment 1. Something in Chapter 6 should be quite helpful for solving it.

7. (2+5+4=12 marks)

One method of hashing with open addressing is to use quadratic probing. In the simplest form of quadratic probing, the ith element of the probe sequence is $h(k, i) = (h(k) + i^2) \bmod M$.

(a) Assume that $h(k) = 0$. Give the probe sequence $\langle h(k, 0), \ldots, h(k, M-1) \rangle$ for $M = 11$ and for $M = 14$. No justification is needed.

(b) Show that this method misses many slots of the hash table. In particular, show that the probe sequence $\langle h(k,0),\ldots,h(k,M{-}1)\rangle$ contains at most $\lceil \frac{M+1}{2} \rceil$ many different values from $\{0,\ldots,M{-}1\}$.

Hint: You should notice in part (a) that many indices appear twice in the probe sequence. Can you detect the pattern in which they repeat?

(c) Argue that if $M \geq 3$ is prime and $\alpha \leq \frac{1}{2}$, then the probe sequence always finds an empty slot.

You are allowed to use modular arithmetic rules without proof, see Appendix B in the textbook for details.