University of Waterloo
CS240 - Spring 2024
Assignment 4

Due Date: Monday, July 15 at 8am

Please read the following link for guidelines on submission:

`https://student.cs.uwaterloo.ca/~cs240/s24/assignments.phtml#guidelines`

Each question must be submitted individually to MarkUs as a PDF with the corresponding file names: a4q1.pdf, a4q2.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.

Late Policy: There's no grace period for this assignment

Question 1 Skip lists [3+3 marks]

Suppose you have a skip list with only three levels. The lower one has $n+2$ entries $-\infty < a_0 < \cdots < a_{n-1} < +\infty$. The middle one has $k+2$ entries, where k is an integer that divides n (so that $n = km$ for some integer m); we assume that (with the exceptions of $\pm\infty$), these entries are evenly spread out, so they correspond to $-\infty, a_0, a_m, a_{2m}, \dots, a_{(k-1)m}, +\infty$. The top level holds $-\infty, +\infty$.

1. What is the worst time for a query? Give a $\Theta(\ )$ expression involving k and n, and justify your answer.
2. Given n, how to choose k so as to minimize the $\Theta(\ )$ bound you just obtained, and what does the worst case become in that case? Give a $\Theta(\ )$ expression in terms of n, and justify your answer. You can assume n is a square.

Question 2 Analysis of self-organizing search [4+4+1+4+4 marks]

In this problem, we analyse the move-to-front strategy for linear search. Suppose we have a linked list with keys $x_1, \dots, x_n$, and that the probability of accessing x_i is p_i; it will be handy later on to assume that $p_i > 0$ for all i.

We consider the move-to-front heuristic. As input, we suppose we are given a linked list with keys $x_1, \dots, x_n$, but we do not know in which order they are.

1. We do not know the order at the beginning, so let us call $X_{i,j}^{(0)}$ the probability that initially x_i is before x_j. Show that after one query, the probability that x_i is before x_j is
$$X_{i,j}^{(1)} = p_i + (1 - p_i - p_j)X_{i,j}^{(0)}$$
and the probability that x_j is before x_i is $1 - X_{i,j}^{(1)}$.

2. Show by induction that after N queries, these probabilities are

$$X_{i,j}^{(N)} = p_i(1 + (1 - p_i - p_j) + (1 - p_i - p_j)^2 + \cdots + (1 - p_i - p_j)^{N-1}) + (1 - p_i - p_j)^N X_{i,j}^{(0)}$$

and $1 - X_{i,j}^{(N)}$.

3. Show that the limit probabilities, for $N \to \infty$, are

$$\frac{p_i}{p_i + p_j} \quad \text{and} \quad \frac{p_j}{p_i + p_j}.$$

You can use the fact that for $0 \leq r < 1$, $1 + r + r^2 + r^3 + \cdots = \frac{1}{1-r}$. In the rest of the problem, we will assume that N is very large, and work with the limit probabilities.

4. Show that for $N \to \infty$ the expected position of x_i in the list is

$$1 + \sum_{j \neq i} \frac{p_j}{p_i + p_j}.$$

Hint: rewrite the number of x_j's before x_i as a sum involving indicator variables. Note: we index positions starting from 1, just as we did in the move-to-front lecture.

5. What is the expected number C_{MTF} of links visited in a search using this heuristic? (do not try to simplify the expression). Compute this value for the frequencies of the example seen in class:

key	A	B	C	D	E
frequency of access	2	8	1	10	5
access-probability	$\frac{2}{26}$	$\frac{8}{26}$	$\frac{1}{26}$	$\frac{10}{26}$	$\frac{5}{26}$

Question 3 Interpolation search [4+4 marks]

1. Suppose that we use interpolation search in an array with entries $A[i] = \alpha i$ for $i = 0, \ldots, n-1$, α positive constant. What is the worst-case runtime for search? Give (and justify) a Θ bound.

2. Suppose that we use interpolation search in an array with entries $A[i] = \sqrt{i}$, for $i = 0, \ldots, n-1$, and that we search for $k = 1$. What is the runtime? Give (and justify) a big-O bound. You can use any result seen in class without reproving them.

Question 4 Tries [3+3 marks]

1. For $n \geq 1$, find the height of the trie that stores the binary representation of all integers in $\{0, \ldots, 2^n - 1\}$ (without unnecessary leading zeros), that is, $0\$, 1\$, 10\$, \ldots$.

2. What is the height of the compressed trie storing these words?

Question 5 Hashing [2+4 marks]

1. Consider a hash table dictionary with universe $U = \{0, 1, 2, \ldots, 24\}$ and size $M = 5$. If items with keys $k = 21, 3, 16, 1$ are inserted in that order, draw the resulting hash table if we resolve collisions using:

    - Linear probing with $h(k) = (k + 1) \mod 5$
    - Cuckoo hashing with $h_1(k) = k \mod 5$ and $h_2(k) = \lfloor k/5 \rfloor$ (use two arrays of size 5)

2. Let S be a set of n keys mapped to a hash table also of size n using chaining. We make the uniform hashing assumption, and we call c_n the expected number of empty slots. Prove that $c_n = n/e + o(n)$, where e is the basis of the natural logarithm.

    You can use the fact that $\lim_{n\to\infty}(1 - 1/n)^n = 1/e$. Hint: use indicator variables $I_0, \ldots, I_{n-1}$, that take values 0 or 1, depending on whether the corresponding entry in the table is empty or not; then, find the expected value of each I_i.

Question 6 One-sided range search in a heap [5 marks]

Heaps are suitable for *one-sided range queries*. Specifically, assume you are given a max-heap H and an integer x, and you are asked to return all keys in H that fall in the range $[x, \infty)$, i.e., all keys that are at least x.

Give an algorithm that answers such a one-sided range query in time that depends only on the number k of keys in the output. Describe your algorithm, justify why it works, and analyze the runtime.