University of Waterloo
CS240 - Winter 2021
Assignment 4

There are written questions in this assignment. The deadline for is Wednesday March 24, 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 24** along with your answers to the assignment; i.e. **read, sign and submit A04-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/~cs240/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: a4q1.pdf, a4q2.pdf, ..., a4q6.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 cs240@uwaterloo.ca and let the ISAs know to look for it.**

Notes:

- Logarithms are in base 2, if not mentioned otherwise.

Problem 1 Interpolation Search [4+4+4=12 marks]

(a) Illustrate interpolation search on array $A = [0, 10, 100, 1000, 10000]$ and search key 10. To illustrate the search, it is enough to show the sequence of values computed for m.

(b) Suppose that we are at some iteration j of interpolation search, and for current values of l, r, it holds that $A[i] = f(i) = ai + b$ for $l \leq i \leq r$, $a \neq 0$, and $a, b \in \mathbb{R}$. Show that interpolation search terminates at iteration j if key k is present in the array A at index between l and r (inclusively). You can assume a > 0.

(c) Change the formula for computing m in the interpolation search so that it stops at iteration j if for the current values of l, r, it holds that $A[i] = f(i) = ai^2 + b$ for $l \leq i \leq r$, $a \neq 0$, and $a, b \in \mathbb{R}$ and if key k is present in the array A at index between l and r (inclusively). You can assume a > 0.

Problem 2 [4+4+4=12 marks]

(a) Draw the standard trie that is obtained after inserting the following keys into an initially empty trie:

$$00\$, 01\$, 011\$, 0111\$, 111\$, 11\$.$$

(b) Repeat part (a) but now construct a compressed trie.

(c) Trace the search for key 000\$ in the trie you have constructed in part (b).

Problem 3 [3+3+3+3+1= 13 marks]

Suppose we have a multiway trie T over alphabet of size m. We will refer to the letters of the alphabet as $a_0, a_1, ..., a_{m-1}$. At each node v of the trie, we use array A_v of size $m+1$ to store the children of node v. For $0 \leq j < m$, $A_v[j]$ contains a pointer to the child v_j of v such that the edge of T connecting v and v_j is labelled with a_j. If no such v_j exists, then $A_v[j] = \emptyset$. $A_v[m]$ stores pointer to the child such that the edge of T connecting v to v_m is labeled with \$. Suppose trie T stores strings in $S = \{(a_0a_1...a_{m-1})^k \mid 1 \leq k \leq n\}$, where notation $(s)^k$ stands for string s repeated k time. For example, $(ab)^3$ is string *ababab*. As usual, when we store strings in the trie, we add the end of string character \$ at the end. Assume that for the uncompressed version of a trie, we do not store strings at the leaves (since they string can be read off from the edge of the trie), but for the compressed trie we do store strings at the leaves. Assume we need $\Theta(l)$ space for a string of length l.

(a) If the trie is uncompressed, what is its height in terms of n? Give the exact number.

(b) If the trie is uncompressed, what is the space requirement as a function of n, m? You can use Θ notation.

(c) If the trie is compressed, what is its height in terms of n? Give the exact number.

(d) If the trie is compressed, what is the space requirement as a function of n, m? You can use Θ notation.

(e) Assume $n > m$. Your result in (b) should be more space efficient than the result in (d). Explain why uncompressed trie is more efficient than the compressed version in this case.

Problem 4 [4+4+4=12 marks]

Consider a hash table dictionary with universe $U = \{0, 1, 2, ..., 25\}$ and size M=7. If items with keys $k = \{21, 7, 0, 9\}$ are inserted in that order, draw the resulting hash table if we resolve collisions using:

(a) Linear probing with $h(k) = (k + 3) \bmod 7$

(b) Double hashing with $h_1(k) = (k + 3) \bmod 7$ and $h_2(k) = 5 -$ (k mod 5)

(c) Cuckoo hashing with $h_1(k) = (k + 3) \bmod 7$ and $h_2(k) = \lfloor \frac{k}{7} \rfloor$

Problem 5 [6 marks]

Write (or explain in English) an algorithm that given an array A of size n storing integers, and a number m returns indices $i \leq j$ such that $\sum_{k=i}^{j} A[k] = m$. If no such indices i, j exist, your algorithm should return null. If there is more than one pair of i, j satisfying the condition, your algorithm can return any such pair. For example, $A = [5, 7, 9, 11, 13, 15]$ and $m = 33$, your algorithm should return $i = 2, j = 4$. Your algorithm must have $O(n)$ expected or average running time.

Problem 6 [5+5+5=15 marks]

In this problem, we allow a quadtree to store arbitrary points, that is points that are not necessarily in general position.

(a) Draw the quadtree corresponding to the set of 2D points:
$S = \{(1, 1), (3, 1), (1, 3), (3, 3), (5, 2), (5, 5), (7, 7)\}$.

(b) Let k be a positive integer and let S be the set of all integer coordinates (i, j) where $0 \leq i < 2^k$ and $0 \leq j < 2^k$, namely $S = \{(i, j) \mid 0 \leq i < 2^k,\ 0 \leq j < 2^k$ and i, j are integers$\}$. What is the height of the quadtree corresponding to S? Express the height as a function of n, where n is the number of points in S. Your expression should be exact, not asymptotic.

(c) Consider the following modification to the quadtree data structure: each leaf is allowed to store up to four points. This means that any region containing 4 or less points is not split any further and is represented as a leaf.

(i) Does the height of the quadtree representing S from part (b) change?

(ii) Let S now be an arbitrary set of n points. Let h be the height of the regular quadtree and h' be the height of the modified quadtree. How large can the difference between h and h' be?