University of Waterloo
CS240 Winter 2025
Assignment 4

Due Date: Tuesday, March 18 at 5:00pm

Please read `https://student.cs.uwaterloo.ca/~cs240/w25/assignments.phtml#guidelines` for guidelines on submission. **Each question must be submitted individually to Crowdmark.** Submit early and often.

Grace period: submissions made before 11:59PM on March 18 will be accepted without penalty. Your last submission will be graded. Please note that submissions made after 11:59PM **will not be graded** and may only be reviewed for feedback.

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

Suppose we have a multiway trie T over an alphabet of size m. We will refer to the letters of the alphabet as $a_1, a_2, ..., a_m$. At each node v of the trie, we use array A_v of size $m + 1$ to store the children of node v. For $1 \leq j \leq m$, $A_v[j]$ contains a pointer to the child v_j of v such that the link of T connecting v and v_j is labelled with a_j. If no such v_j exists, then $A_v[j] = \emptyset$. $A_v[0]$ stores pointer to the child such that the link of T connecting v to v_m is labeled with \$. Suppose trie T stores strings in $S = \{(a_1a_2...a_m)^k \mid 1 \leq k \leq n\}$, where notation $(s)^k$ stands for string s repeated k times. 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 the string can be read off from the link 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 T is uncompressed, what is its height in terms of n, m? Give the exact number.

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

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

(d) If the trie T 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 the uncompressed trie is more efficient than the compressed version in this case.

Question 2 [2+2+3+3=10 marks]

Consider a hash dictionary with table of size $M = 10$.
Suppose items with keys $k = \{4371, 1323, 6173, 4199, 4344, 1679, 1989\}$ are inserted in that order using hash function $h_1(k) = k (\mathrm{mod}\ 10)$. Draw the resulting hash table if we resolve collisions using

a) Chaining

b) Linear probing

c) Double hashing with the secondary hash function $h_2(x) = \lfloor \frac{x}{1000} \rfloor$.

d) Cuckoo hashing with the secondary hash function $h_2(x) = \lfloor \frac{x}{1000} \rfloor$.

Question 3 [1+2+1+3=7 marks]

Suppose we have a hash table with chaining of size M and we insert n items into the hash table. Assume uniform hashing.

a) What is the probability that the first and the second items we insert end up in the first bucket?

b) What is the probability that all n items end up in the same bucket?

c) What is the probability that one bucket contains 2 or more items if $n > M$?

d) What is the probability that one bucket contains 2 or more items if $n = M$?

Question 4 [6 marks]

Design an algorithm that given an array A of size n storing integers and a number m, returns indices i and j, $i \leq j$ 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, given $A = [5, 7, 9, 11, 13, 15]$ and $m = 33$, your algorithm should return $i = 2, j = 4$. Your algorithm must have $O(n)$ expected running time. You can explain your algorithm in English or write pseudocode.

Question 5 [4+4=8 marks]

a) Draw the Quadtree for the following set of 2D points S:

$$S = \{(3, 5), (7, 8), (6, 2), (8, 0), (0, 3), (4, 6), (2, 9), (9, 1)\}.$$

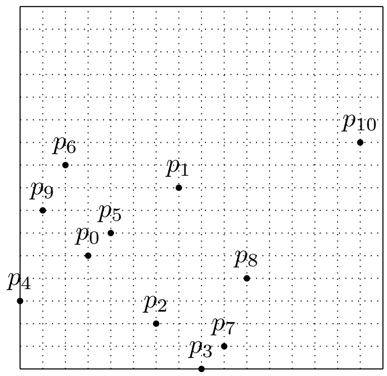
b) Suppose we have a modified Quadtree available where each leaf is allowed to store up to 4 points. Let T be a regular Quadtree of height $h = 100$ and let T' be a modified Quadtree. Both T and T' store the same points. What is the largest and smallest possible height of T'? Explain.

Question 6 [4+4+(2+2+2)=14 marks]

Consider the following set of points in two dimensions:

$$S = \{(3,5),(7,8),(6,2),(8,0),(0,3),(4,6),(2,9),(9,1),(10,4),(1,7),(15,10)\}.$$

a) Draw the kd-tree and its corresponding plane partition diagram.



b) Show how a search for the points in $R = [1.6 \le x < 6.5, 4.5 \le y < 11.5]$ would proceed. More specifically, describe the 'colour' (blue, green, red) of nodes as done in the Course Slides for Module 8, Slide 21. Note this is not referencing Professor Olga's slides. You can directly colour the nodes in the picture you drew in (a).

c) The goal of this exercise is to show that splitting a kd-tree both vertically and horizontally is crucial for efficient search. Consider a modified kd-tree which we will call a degenerate kd-tree. A degenerate kd-tree uses only vertical splits, never horizontal. For example, let $S = \{(1,2),(2,3),(3,4),(4,5)\}$. Then a degenerate kd-tree first splits the space into two regions with line $x = 3$, with points $(1,2),(2,3)$ on the left and $(3,4),(4,5)$ on the right. Then the left region is split with a line $x = 2$, and the right region with the line $x = 4$. Let $S = \{(i,i) \mid i \in \{1,...,n\}\}$. Let T be a degenerate tree on S. You can assume n is a power of 2.

i) What is the height of T? Explain your answer.

ii) How many nodes does T have? Explain your answer.

iii) Give an example of a range search on S such that it visits all the nodes of the degenerate kd-tree yet the search result is empty. Explain your answer.