University of Waterloo
CS240, Fall 2018
Assignment 4

Due Date: Wednesday, November 14, at 5pm

Please read `http://www.student.cs.uwaterloo.ca/~cs240/f18/guidelines.pdf` for guidelines on submission. Problems 1—3 are written problems; submit your solutions electronically as a PDF file with name `a04wp.pdf` using MarkUs. We will also accept individual question files named `a04q1w.pdf`, `a04q2w.pdf`, `a04q3w.pdf` and `a04q4w.pdf`. Problem 5 is a programming problem; submit your solution to 5 electronically as a file named `kdpartition.cpp`.

There are 55 possible marks available. The assignment will be marked out of 50.

Problem 1 Hashing [3+3+3+3+2=14 marks]

Consider a hash table dictionary with universe $U = \{0, 1, 2, \ldots, 25\}$ and size $M = 5$. If items with keys $k = 17, 10, 20, 13$ are inserted in that order, draw the resulting hash table if we resolve collisions using:

a) Chaining with $h(k) = (k + 2) \bmod 5$.

b) Linear probing with $h(k) = (k + 2) \bmod 5$

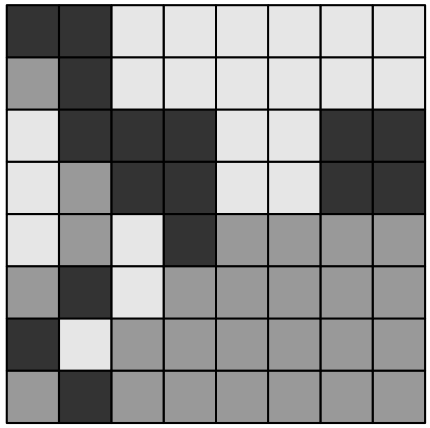
c) Double hashing with $h_1(k) = (k + 2) \bmod 5$ and $h_2(k) = 1 + (k \bmod 4)$.

d) Cuckoo hashing with $h_1(k) = (k + 2) \bmod 5$ and $h_2(k) = \lfloor k/5 \rfloor$.

e) Identify a serious problem with the choice of hash functions in part (d).

Problem 2 Quad Trees [6+3=9 marks]

a) One of the applications of quad trees is for image compression. An image (picture) is recursively divided into quadrants until the entire quadrant is only one colour. Using this rule, draw the quad tree of the following image. Use the convention that each internal node of a quad tree has exactly four children, corresponding to regions R_{NE}, R_{NW}, R_{SW} and R_{SE}, in that order. There are only three colours (shades of grey). For the leaves of the quad tree, use 1 to denote the lightest shade, 2 for the middle shade and 3 for the darkest shade of grey. The bounding boxes in the internal nodes do not need to be indicated.



b) For an integer $k \geq 2$, consider a quad tree with bounding box $[0, 2^k) \times [0, 2^k)$ storing three points $\{(0,0), (2,2), (2^k - 1, 2^k - 1)\}$. For $k \geq 2$, determine the height of the quadtree.

Problem 3 Range Trees [5+5+5=15 marks]

a) Assume that we have a set of n numbers (not necessarily integers) and we are interested only in the number of points that lie in a range rather than in reporting all of them. Describe how a 1-dimensional range tree (i.e., a balanced binary search tree) can be modified such that a range counting query can be performed in $O(\log n)$ time (independent of k, the number of nodes). Provide the range counting query and justification of its runtime.

b) Now consider the 2-dimensional-case: We have a set of n 2-dimensional points. Given a query rectangle R, we want to find the number of points that lie in R. Preprocess the n points (by building an appropriate range-tree based data structure) such that you can answer any of these counting queries in time $O((\log n)^2)$. Provide the range counting query and justification of its runtime.

c) Suppose a two dimensional range tree data structure stores n points, and that the x-BST is perfect, i.e., every level is completely filled. Derive an exact closed form formula in terms of n for the sum of the number of nodes in the x-BST plus the total number of nodes in all y-BSTs.

Problem 4 Covered Windows [7 marks]

We are given a set $\mathcal{W}$ of n of windows on the computer screen S (i.e., axis-parallel rectangles in the plane). A new window pops up and we wish to find all windows in $\mathcal{W}$ that are completely covered by the new window. Give an algorithm that finds all these windows in $O((\log n)^c + s)$ time, where s is the number of windows that are found, and $c \geq 1$ is a constant.

Formally, each window is described via a 4-tuple (x_ℓ, x_r, y_b, y_t) with $x_\ell < x_r$ and $y_b < b_t$, which corresponds to the rectangle $[x_\ell, x_r] \times [y_b, y_t]$. If the new window is $W' = (x'_\ell, x'_r, y'_b, y'_t)$, your query should return in $O((\log n)^c)$ time all those windows $W = (x_\ell, x_r, y_b, y_t)$ in $\mathcal{W}$ that satisfy $[x_\ell, x_r] \subseteq [x'_\ell, x'_r]$ and $[y_b, y_t] \subseteq [y'_b, y'_t]$.

To make this possible, you will need to assume that $\mathcal{W}$ is stored in a suitable data structure. Describe what you are using. This data structure should take at most $O(n \log^c n)$ space, and one should be able to build it in $O(n \log^c n)$ time.

Problem 5 *kd*-Tree Construction [10 marks]

Implement an algorithm to construct a *kd*-tree for dimension 2. Your algorithm should read $2n + 1$ integers from standard input, separated by white space. The first integer is the number of points. The remaining $2n$ integers are the points themselves, according to their x and y coordinates.

Actually, your program does not need to construct the tree, but rather should just print to standard output the n points in the order they are visited during an **in-order traversal** of the *kd*-tree. Thus, your output should consist of $2n$ integers separated by whitespace.

- You may use any standard library function, for example to sort or partition an array.
- The tree you construct should correspond to the tree produced by the recipe on Slide 15 of Module 8.
- You may assue that the input points are in generic position.
- Your answer to this question will be autotested.
- Here are two example input and output files.

  - Input: `4 3 4 2 2 0 1 1 3`
    Output: `0 1 1 3 2 2 3 4`
  - Input: `5 0 0 1 2 2 1 3 3 4 4`
    Output: `0 0 1 2 2 1 3 3 4 4`