University of Waterloo
CS240, Fall 2015
Assignment 4

Due Date: Wednesday, November 18, at 5pm

Please read `http://www.student.cs.uwaterloo.ca/~cs240/f15/guidelines.pdf` for guidelines on submission. Problems 1—4 and 5(b) 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`, `a04q4w.pdf` and `a04q5w.pdf`. Problem 5(a) is a programming problem; submit your solution to 5(a) electronically as a file named `kdpartition.cpp`.

There are 74 possible marks available. The assignment will be marked out of 70.

Problem 1 Hashing [3+3+3+3=12 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 $h_2(k) = \lfloor k/5 \rfloor$.

Problem 2 Extendible Hashing [12 marks]

Suppose we have an extendible hashing scheme with block size $S = 3$ and parameter $L = 5$. The universe of keys is non-negative integers with at most 8 bits, $U = \{0, 1, \ldots, 255\}$, and the hash function is

$$h(k) = \lfloor k/16 \rfloor + (k \bmod 16).$$

The dictionary is initially empty, with a single block B, pointed to by the single entry in the directory, which has initial order $d = 0$.

Show the result of inserting the following **keys** into the dictionary, in order. In each block location, you should write the key and the corresponding hash value. Indicate the order d and the local depth k_B of each block.

You don't need to draw out every single step, but you must indicate clearly every time a *block split* or *directory grow* occurs, and draw the resulting structure immediately following each such operation, and at the end of all the insertions.

The sequence of keys to insert is: 191, 142, 192, 248, 217, 95

Problem 3 Skip Lists [6+6+6=18 marks]

a) Starting with an empty skip list, insert the seven keys $54, 15, 51, 53, 47, 68, 36$. Draw your final answer as on Slide 7 in Module 6. Use the following coin tosses to determine the heights of towers (note, not every toss is necessarily used):

$$T, T, H, H, T, H, T, H, H, T, H, H, T, T, H, T, H, H, T, T, H, H, H, T, \ldots$$

b) The worst case time for searching in a singly linked list is $\Theta(n)$. Now consider a variation of a skip list which has fixed height $h = 3$ even though n can become arbitrarily large. Level S_0 contains the keys $-\infty, k_1, k_2, \ldots, k_n, \infty$. Level S_3 contains only $-\infty$ and ∞. Describe subsets of keys that should be included in levels S_1 and S_2 so that searching in the skip list has worst case cost $\Theta(n^{1/3})$. Provide justification for the runtime of your skip list.

c) Consider a skip list in which we build new towers with probability 1/4.

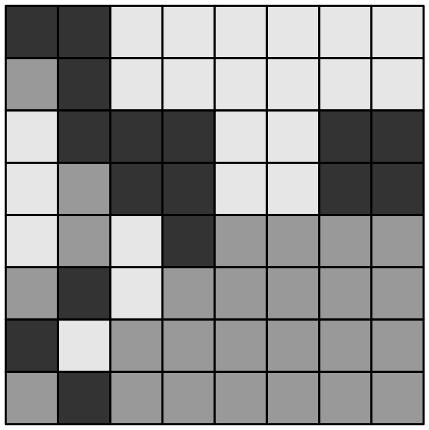
> When adding an element to the skip list, we flip two coins at the same time, until we see at least one tail. The number of times we toss both coins and obtain two heads is the height of the tower.

Using the probability for tower heights described in the above quote, derive the expected height of any single tower (not the entire skip list).

Problem 4 Quad Trees [6+6=12 marks]

For both parts of this question, use the convention that each internal node of a quad tree has exactly four children, corresponding to regions NW, NE, SE and SW, in that order.

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. 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.



b) Give three 2-dimensional points such that the corresponding quad tree has height exactly 7. Give the (x,y) coordinates of the three points and show the quad tree. (Do not give the plane partition.)

Problem 5 *kd*-Tree Construction [10+10=20 marks]

a) 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 an array.
- Your output must correspond to the tree produced by the recipe on Slide 13 of Module 7.
- Your answer to this question will be autotested. However, for the next part question regarding efficiency the graders will examine the code you have submitted.

Here is an example input and output file.

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

b) Analyse the (worst case) running time of the algorithm you have implemented. The graders must be able to quickly match your analysis with the code you have submitted. If your analysis is correct you will receive for this part question

- full marks for a linearithmic running time bound;
- half marks for a subquadratic running time bound.
  (Note: An algorithm is said to be linearithmic if its running time is $O(n \log n)$ and an algorithm is said to run in subquadratic time if its running time is $o(n^2)$).