CSCI 3104 **Instructors Buxton & Winchell**
Problem Set 3 **Spring 2018, CU-Boulder**

1. (10 pts total) For parts (1a) and (1b), justify your answers in terms of deterministic QuickSort, and for part (1c), refer to Randomized QuickSort. In both cases, refer to the versions of the algorithms given in lecture (you can refer to the moodle lecture notes).

   (a) What is the asymptotic running time of QuickSort when every element of the input A is identical, i.e., for $1 \leq i, j \leq n$, $A[i] = A[j]$?

   (b) Let the input array $A = [9, 7, 5, 11, 12, 2, 14, 3, 10, 6]$. What is the number of times a comparison is made to the element with value 3?

   (c) How many calls are made to `random-int` in (i) the worst case and (ii) the best case? Give your answers in asymptotic notation.

2. (20 pts total) Use the Master Theorem to solve the following recurrence relations. For each recurrence, either give the asympotic solution using the Master Theorem (state which case), or else state the Master Theorem doesn't apply.

   (a) $T(n) = T(\frac{3n}{4}) + 2$

   (b) $T(n) = 3T(\frac{n}{4}) + nlgn$

   (c) $T(n) = 8T(\frac{n}{3}) + 2^n$

   (d) $T(n) = T(\frac{n}{2}) + T(\frac{n}{4}) + n^2$

   (e) $T(n) = 100T(\frac{n}{42}) + \lg n$

3. (30 pts total) Thormund the cleric has created n vials of holy water for an upcoming quest into a graveyard – little does he know, Grog has accidentally spilled ale into some of these vials. Harry saw Grog's mistake and tells Thormund he needs to identify which of vials of holy water are tainted. Together with Harry, Thormund has constructed a strange contraption that fits over two vials at a time to perform a test. When the contraption is activated, each vial glows one of two colors depending on whether the *other* vial is tainted or not. An untainted vial always glows correctly according to whether the other vial is accurate or not, but the glow of a tainted vial cannot be trusted. You quickly notice that there are four possible test outcomes:

vial i glows	vial j glows		
red	red	$\Longrightarrow$	at least one is tainted
red	green	$\Longrightarrow$	at least one is tainted
green	red	$\Longrightarrow$	at least one is tainted
green	green	$\Longrightarrow$	both are untainted, or both tainted

(a) Prove that if $n/2$ or more vials are tainted, Thormund cannot necessarily determine which vials are tainted using any strategy based on this kind of pairwise test. Assume a worst-case scenario in which the tainted vials are used, meaning the tainted vials always show the opposite color of the other vial.

(b) Consider the problem of finding a single good vial from among the n vials, and suppose Thormund knows that more than $n/2$ of the vials are untainted, but not which ones. Prove that $\lfloor n/2 \rfloor$ pairwise tests are sufficient to reduce the problem to one of nearly half the size.

(c) Now, under the same assumptions as part (3b), prove that all of the untainted vials can be identified with $\Theta(n)$ pairwise tests. Give and solve the recurrence that describes the number of tests.

4. (20 pts total) Harry needs your help breaking into a dwarven lock box. The lock box projects an array A consisting of n integers $A[1], A[2], \ldots, A[n]$ and has you enter in a two-dimensional $n \times n$ array B – to open the box – in which $B[i,j]$ (for $i < j$) contains the sum of array elements $A[i]$ through $A[j]$, i.e., $B[i,j] = A[i]+A[i+1]+\cdots+A[j]$. (The value of array element $B[i,j]$ is left unspecified whenever $i \geq j$, so it doesn't matter what the output is for these values.) Normally Harry would do the computations himself, but the lock box changes the input array after a few minutes, thus Harry needs a fast way to solve this problem.

Harry suggests the following simple algorithm to solve this problem:

```
dwarvenLockBox(A) {
   for i = 1 to n {
      for j = i+1 to n {
         s = sum(A[i..j])        // look very closely here
         B[i,j] = s
}}}
```

(a) For some function g that you should choose, give a bound of the form $\Omega(g(n))$ on the running time of this algorithm on an input of size n (i.e., a bound on the number of operations performed by the algorithm).

(b) For this same function g, show that the running time of the algorithm on an input of size n is also $O(g(n))$. (This shows an asymptotically tight bound of $\Theta(g(n))$ on the running time.)

(c) Although Harry's algorithm is a natural way to solve the problem—after all, it just iterates through the relevant elements of B, filling in a value for each—it contains some highly unnecessary sources of inefficiency. Give an algorithm that solves this problem in time $O(g(n)/n)$ (asymptotically faster) and prove its correctness.

5. (20 pts total) Consider the following strategy for choosing a pivot element for the `Partition` subroutine of QuickSort, applied to an array A.

   - Let n be the number of elements of the array A.
   - If $n \leq 24$, perform an Insertion Sort of A and return.
   - Otherwise:
     - Choose $2\lfloor n^{(1/2)} \rfloor$ elements at random from n; let S be the new list with the chosen elements.
     - Sort the list S using Insertion Sort and use the median m of S as a pivot element.
     - Partition using m as a pivot.
     - Carry out QuickSort recursively on the two parts.

   (a) How much time does it take to sort S and find its median? Give a Θ bound.

   (b) If the element m obtained as the median of S is used as the pivot, what can we say about the sizes of the two partitions of the array A?

   (c) Write a recurrence relation for the worst case running time of QuickSort with this pivoting strategy.