CSCI 3104
Spring 2018
Problem Set 3

Allison, George
02/17

February 8, 2018

1. a) The asymptotic running time of QuickSort is $O(n^2)$(worst case) when every element is identical.

b) The value 3 is compared four times.

9 7 5 11 12 2 14 3 10 6 Array A
6 7 5 2 3 9 14 12 10 11 Sorted pivot, one comparison made
6 5 2 3 7 left array sort w/ pivot as 6; one comparison made
3 5 2 6 7 left array sort w/ pivot as 3; one comparison made
2 3 5 left array sort w/ pivot as 2; one comparison made
2 3 after this point there will be no more comparisons made with three.

c) In the worst case scenerio there are $O(n)$ random-int calls, and in the best case there are $O(nlg(n))$ random-int calls.

2. a) If more than half (n/2) of the crystal balls are inaccurate then the pairwise tests will not be useful because every test needs to be used multiple times to determine which crystal balls are accurate. If the number of inaccurate crystal balls is more than the number of accurate crystal balls it will be impossible to distinguish the accurate from the inaccurate with any oairwise tests. Once one crystal ball is known to be accurate it can be used to test other crystal balls to determine their accuracy, but with more than n/2 inaccurate crystal balls it would not be possible to obtain the first one.

b) Trelawney would need to test each pair of crystal balls. The pairs where at least one is inaccurate (red-red, red-green, green-red) should be discarded and the pairs where both are either accurate or inaccurate (green-green) are kept. By switching around the pairs of crystal balls Trelawney can narrow down the results until one is proven to be accurate. The proven-accurate ball can be used to test the rest of the crystal balls for accuracy.

c) The process of searching for one accurate crystal ball via pairwise tests can be represented as $T(n) \leq T(\lceil n/2 \rceil) + \lfloor n/2 \rfloor$ because the process of elimination will at least check all of the accurate crystal balls but is limited by the pairs of crystal balls that are removed. After obtaining a proven-accurate crystal ball the process of checking it against all the others is $\Theta(n)$. The solution to the complete recurrence is $T(n) = O(n)$

3. a) The two loops, each iterating from 0 to n, combined with the sum function results in $\Omega(n^3)$. The algorithm can be reprsesnted as $\frac{n}{2}$(counting from i to j) * $\frac{3n}{4}$(performing operations on each pair) $= \frac{n^3}{32}$ which proves the running time $\Theta(n^3)$

b) The running time of the algorithm on an n-sized input is also $O(n^3)$ because the number of iterations from i to j will always occur in the two for-loops (resulting in $O(n^2)$ complexity) with the sum function resulting in a constant number of iterations, at most being n iterations, so the algorithm will never grow faster than $O(n^3)$

c) An algorithm that can do the numerical computation while it is iterating through the values would have a better time complexity ($\Theta(n^2)$ than the orginal algorithm. I don't know how to write that sort of algorithm out.

4) i) The calls to hagHelp have two parameters, while that function takes three paramenters.
ii) I don't know
iii) I don't know