1. (5) Provide the runtime for counting sort. Identify the source of each parameter

2. (10) Can the worst case performance be avoided in quicksort? Why or why not?

3. (5) What is the purpose of the max-heapify routine? What is its runtime?

4. (10) How does Strassens achieve a better runtime over the traditional divide and conquer matrix multiplication? Provide support for your answer
5. (10) Given an array containing 9, 6, 5, 14, 10, 2, 11, 3, 8 provide the maximum left subarray, maximum right subarray and maximum crossing subarray for each recursion.

6. (5) Add the following values in order to the existing max heap below 7, 12, 2, 13, 11
7. (15) Solve each recurrence using the master method, and write the solution as a simplest $\Theta$ function of $n$. If it is not solvable using the Master method, indicate so.

   a. $T(n) = 9 \, T(n/3) + n^3$

   b. $T(n) = 8 \, T(n/2 + 1) + n^4$

   c. $T(n) = 6 \, T(n/6) + n$

8. (5) For the following $T(n)$, provide a $\Theta$, $\Omega$, $\omega$, $O$ and $o$ bound. Provide support for each answer.

   $T(n) = n^2 + 6n$

9. (10) Provide a code snippet that is $\Theta(\sqrt{n})$ with a single outer loop
10. (10) Provide the first 3 levels of the recursion tree for the recurrence \( T(n) = 2T(n/3) + T(n/2) + n^2 \)

11. (5) Given a radix sort of UPC codes of 48 bits, determine a recommended number of passes if there are 65,000 values. Provide support for your answer. How will this change if the UPC codes have 32 bits?

12. (10) Provide the intermediate array contents after the first two passes (should touch each number twice) of quick sort

61, 75, 96, 82, 96, 55, 119, 101, 125, 126, 186, 58