CS/DSA ALGORITHM ANALYSIS
HW5 PRACTICE PROBLEMS, Sept 19, 2019.

1. Find the upper and a lower bound on $T(n) = \frac{n^3}{1000} - 100n^2 - 100n + 3$.

2. Describe a $O(nlogn)$ algorithm that, given a set S of n integers and another integer x, determine whether or not there exist two elements in S whose sum is x.

3 What is an inversion in a permutation ? Find the number of inversions in

$$P = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 5 & 4 & 3 & 2 & 1 \end{bmatrix}$$

4. Show that $(n+a)^b = \theta(n^b)$, where a and b are real and positive.

5. Is $2^{n+1} = O(2^n)$? Is $2^{2n} = O(2^n)$?

6. Is the function $\lceil log_e n \rceil!$ polynomially bounded? Is the function $\lceil log_e log_e n \rceil!$ polynomially bounded?

7. Which is asymptotically larger: $log(log^* n)$ or $log^*(logn)$?

8. Solve the Fibonacci recurrence $F_k = F_{k-1} + F_{k-2}$ when $F_0 = 0$ and $F_1 = 1$. Compute F_{10}, F_{50}. F_{100}

9. Given a constant $c > 0$, define the iterated function f_c^* by $f_c^*(n) = min\{k \geq 0 : f^k(n) \leq c\}$ where $f^{(1)}(n) = f(n)$, $f^{(2)}(n) = f(f(n))$ and so on. Compute the value of f_c^* when

a. f(n)= n-1 and c=0
b. f(n)= $\frac{n}{2}$ and c=1
c. f(n) = logn and c=1.

10. Solve the following recurrences using substitution method discussed in Appendix:

a. T(n)= T(n-1) + $\frac{1}{n}$, T(1)= 1
b. T(n)= $\sqrt{n}T(\sqrt{n})$ + n, $n = 2^{2^k}$, T(2)= 1
c. T(n)= T(n-1) + logn, T(1)= 1

d. $T(n) = 7T(\frac{n}{2}) + 18(\frac{n}{2})^2$, $n = 2^k$, T(1)=1.

11. Prove that the cost of building the initial heap is O(n).

12. Derive the recurrence for the average case complexity of Quick-Sort.
Solve: $A(n) = \sum_{k=2}^{n-1} A(k) + (n-1)$ and A(1)= 0, A(2)= 1.

13. What is Stirlings formula? Using it compute an approximation to 10!, 20! and compare with exact values.

14. Describe an algorithm for finding the median of a set of n numbers?

- Reading Assignment: Chapter 8 on "Sorting in Linear time".