Sam Hopkins
Hw 4: 2.4, 2.5 (a-e, only bounds using master theorm)
2.17

2.4) Algorithm A: $a=5,\ b=2,\ d=1$

$\frac{5}{2^1} = O(n^{\log_2 5})$

$\frac{a}{b^d}\begin{cases} <1 = O(n^d) \\ =1 = O(n^d \log n) \\ >1 = O(n^{\log_b a}) \end{cases}$

Algorithm B: $a=2,\ b=n-1,\ d=K$ $2T(n-1)+K \to O(2^{n-1}) \to O(2^n)$

Alg. C: $a=9,\ b=3,\ d=O(n^2)(2)$

$\frac{9}{3^2} \to 1 = O(n^2 \log n)$

Algorithm C is the fastest time

2.5) a) $a=2,\ b=3,\ d=0$ $\frac{2}{3^0} = \frac{2}{1} = O(n^{\log_3 2})$

b) $a=5,\ b=4,\ d=1$ $\frac{5}{4^1} = \frac{5}{4} > 1 = O(n^{\log_4 5})$

c) $a=7,\ b=7,\ d=1$ $\frac{7}{7^1} = 1 \to O(n \log n)$

d) $a=9,\ b=3,\ d=2$ $\frac{9}{3^2} = 1 \to O(n^2 \log n)$

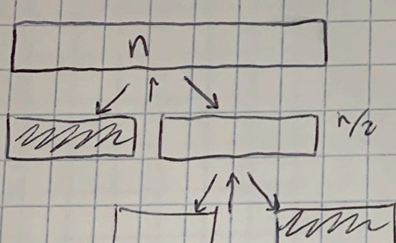
e) $a=8,\ b=2,\ d=3$ $\frac{8}{2^3} = 1 \to O(n^3 \log n)$

2.17) $aT(n/b) + O(n^d)$ $\frac{a}{b^d} = 1 \to \frac{a}{b}$ $\log n$ $d=0$

$d=0,\ a=b$ so $a=1$

binary search

$T(n) = T\left(\frac{n}{2}\right) + 1$



because we throw out $n/2$ indexes each time, b is 2, and $a=1$, with recombining being constant.

So by master theorm, $a=1,\ b=2,\ d=0,\ \frac{a}{b^d} = \frac{1}{2^0} = \boxed{O(\log n)}$

$O(n^0 \log n)$

We find the middle and compare, if greater than target, we trash the right side, and continue with left, or if less, go with right side continue until only one left, and compare if is target or not.