Derrick DuBose
#1536268

7.2-1) Prove $T(n) = T(n-1) + \Theta(n) = \Theta(n^2)$

$$T(n-1) \leq (n-1)^2$$

$$T(n) \leq (n-1)^2 + \Theta(n)$$

$$\leq n^2 - 2n + 1 + \Theta(n)$$

$$T(n) = \Theta(n^2)$$

removed lower degree terms and coeffients

7.2-2) If every element is the same then i will get incremented til i is = r-1 which means the partition will be $T(n-1)$ conquering takes $\Theta(n)$ time

so $T(n) = T(n-1) + \Theta(n)$

$$\boxed{T(n) = \Theta(n^2)}$$

7.2-3) Every time we go through the array the pivot is the smallest element in the unsorted subarray so the partition will be $T(n-1)$ and conquering still takes $\Theta(n)$ time to do.

so $T(n) = T(n-1) + \Theta(n)$

$$\boxed{T(n) = \Theta(n^2)}$$

7.2-4) If the check numbers are already in sorted order then insertion sort will take $\Theta(n)$ and the Quick sort $\Theta(n^2)$ to sort an already sorted array. If the checknumbers are in increasing order as you use them then there won't be many elements to sort. Since Quicksort's avg case time of $\Theta(n \lg n)$ occurs when the numbers are fairly mixed up to let the partition divide the problem in 2 fairly equal subproblems in size. Since Bank checks are already fairly in sorted then insertion sort would be a better procedure.

7.2-5) if $0<a\leq 1/2$ then $a\leq 1-a$ which means a is the smaller or equal to problem size $(1-a)$.

If a problem of size n is reduced then the minimum depth of a leaf is an and the maximum depth of a leaf is $(1-a)n$.

The length of each path is found by dividing n from $1/a$ and $1/(1-a)$ until the base case $n=1$ is met

$\uparrow$ min $\quad$ $\uparrow$ max

length of min $= \log_{1/a} n = \log_{a^{-1}} n = \frac{\log n}{\log a^{-1}} = \boxed{\frac{\log n}{-\log a}}$

length of max $= \log_{1/(1-a)} n = \frac{\log n}{\log (1-a)^{-1}} = \boxed{\frac{\log n}{-\log(1-a)}}$