Jonathan Ameri
CSE 3500 HW 7

Submit a short report containing the following.
(40 points) Provide the table comparing the two versions by listing the running time for each data size.

Input size	Version 1 (last element)	Version 2 (med of 3 random)
N = 10	0.0251	0.0313
N = 100	1.5573	0.3088
N = 200	5.4788	0.9123
N = 500	35.6595	1.8977
N = 1000	137.2155	3.8428
N = 1500	315.1822	5.812
N = 2000	545.6853	8.177

(Note: Time is in ms)

(30 point) Draw conclusion on why choice of pivot matters.
Quicksort has a worst case scenario of O(n^2). This only happens when the pivot picked with each recursive call is the worst pivot possible. For our regular quicksort algorithm, a sorted list is the worst case scenario. When we implement quicksort where we pick each pivot based on 3 random elements, we are more likely to get a better pivot with each recursive call. Randomly picking indices from the list reduces the chance of getting a worst-case scenario to practically 0, especially for larger lists. It also guarantees that there is no predetermined list you can input to get a worst case scenario. Picking 3 random elements and using the median as the pivot guarantees that the runtime will be around O(nlogn) no matter the input.

(30 points) Code Attach the source code of your implementation. If it is short enough, you may simply include your code as part of your report