

Lab 5

1. Show all steps of QuickSort in sorting the array [1, 6, 2, 4, 3, 5]. Use leftmost values as pivots at each step.
2. In our average case analysis of QuickSort, we defined a *good self-call* to be one in which the pivot x is chosen so that number of elements $< x$ is less than $3n/4$, and also the number of elements $> x$ is less than $3n/4$. We call an x with these properties a *good pivot*. When n is a power of 2, it is not hard to see that at least half of the elements in an n -element array could be used as a good pivot (exactly half if there are no duplicates). For this exercise, you will verify this property for the array $A = [5, 1, 4, 3, 6, 2, 7, 1, 3]$ (here, $n = 9$). Note: For this analysis, use the version of QuickSort in which partitioning produces 3 subsequences L, E, R of the input sequence S .
 - a. Which x in A are good pivots? In other words, which values x in A satisfy:
 - i. the number of elements $< x$ is less than $3n/4$, and also
 - ii. the number of elements $> x$ is less than $3n/4$
 - b. Is it true that at least half the elements of A are good pivots?
3. Give an $o(n)$ (“little-oh”) algorithm for determining whether a sorted array A of distinct integers contains an element m for which $A[m] = m$. You must also provide a proof that your algorithm runs in $o(n)$ time.
4. Devise a pivot-selection strategy for QuickSort that will guarantee that your new QuickSort has a worst-case running time of $O(n \log n)$.
5. Show the steps performed by QuickSelect as it attempts to find the median of the array [1, 12, 8, 7, -2, -3, 6]. (The median is the element that is less than or equal to $n/2$ of the elements in the array. Since n is odd in this case, it is the element whose position lies exactly in the middle. Hint: The median is 6.) For pivots, always use the leftmost element of the current array.