Exercises 18 : quicksort and select (deterministic)

Questions

1. In the lecture, when I presented linear time selection algorithm, I partitioned the list into two lists whose elements were less than the pivot (l1) and greater than or equal to the pivot (l2), respectively. However, I neglected a problem in doing so, namely that when elements are allowed to be equal (from an ordering perspective), you are no longer guaranteed to have a 30/70 split. Explain the problem and find a simple way to correct the problem.

2. In class I presented the median of median of 5's algorithm for selecting the ith largest element in an ordered set. Why did I choose 5? What happens if you try the same idea using the median of median of 3's ? That is, you partition the input into triplets, choose the median of each triplet, find the median of these median-of-3's, and then use that median as the pivot to partition the original list. Give a recurrence for the median-of-median-of-3's method.

3. Show that for the median-of-median-of-3's method, the recurrence doesn't give a proof that t(n) is O(n). That is, if you try to do the same proof as I gave in the lecture, the proof doesn't work. (This isn't quite enough to show that t(n) really isn't O(n) for the median-of-3's, but its all you probably want right now.)

Answers

1. First take the extreme case that all the elements are equal. What happens? The pivot will partition the list into two lists: one having all the elements less than the pivot. This list will be empty. Uh-oh. The other set will have all the elements greater than or equal to the pivot. This list will have n-1 elements, all equal to the pivot.
   Notice this problem doesn't just happen when *all* the elements are equal. It happens more generally when the median-of-medians that you select as the pivot is equal to many other elements. In this case, we are no longer certain that less than 3n/10 elements are strictly less than this chosen pivot.

   Can we correct the problem? Yes we can! Once we've chosen the pivot, and partitioned into lists l1 and l2, we check the size of l1. If it is less than n/3, then there are elements in the l2 list that are equal to the pivot. We then move elements from l2 to l1 to make sure the two lists are better balanced, i.e. at least a 30%:70% split. We can do that in O(n) time so it doesn't cost us anything extra in the O() sense, beyond the partitioning

2. There are two recursive calls to select. The first is on the n/3 median-of-3's and it selects the median of these. This leaves n/3/2 = n/6 median-of-3's that are less than the pivot and n/6 median-of-3's that are greater than the pivot. (Note that there are off-by-one and roundoff issues which I am totally ignoring here. These don't matter for the analysis.) For each of the n/6 median of 3's that are less than the pivot, we have another element that is less than the pivot, namely the 1-of-3 that is less than each median-of-3. So, we have n/6 + n/6 = n/3 elements that we know are less than the pivot. By the same argument, we have n/3 elements that we know are greater than the pivot. Thus, the pivot is guaranteed to partition the original list of size n into two lists that are both of size at least n/3 and at most 2*n/3. Thus, in the worst case, the second recursive call of select is on a list of size at most 2*n/3.
   Since the two recursive calls of select are on sizes n/3 and 2*n/3, and cn operations are needed e.g. to find the n/3 median-of-3s, the recurrence is:
   t(n) <= t(n/3) + t(2*n/3) + c n.

3. Suppose there exists a β such that t(k) <= β k for all k < n. Next, substitute this inequality into the recurrence from the previous question:
   t(n) <= β n/3 + 2β n/3 + c n
   = β n + c n
   = (β + c) n.
   But this is too weak a bound. There is no way we can choose β and c to be sure that t(n) <= β n. The basic problem is that n/3 + 2*n/3 = n, whereas with the median of median of 5's we had n/5 + 7n/10 = 9/10 n < n. It was the fact that 9/10 < 1 that allowed us to obtain the O(n) in that case.
   If you want to discuss this further and show that t(n) is NOT O(n) for the median of median of 3's, let me know.