

Due at the start of class Monday, July 6, 2015.

Problem 1.

- (a) Illustrate the operation of radix sort on the following list of English words: RUTS, TOPS, SUNS, SPOT, TONS, OPTS, TORS, SOTS, ROOT, OUTS, SUPS, PUTT
- (b) Write an English sentence using both “tor” and “sot” (that indicates you understand the meanings of both words).

Problem 2. We will try to get a better analysis of the probabilistic algorithm for selection. In class we made the conservative assumption that after partition the element being selected for was always on the larger side. For this problem we will assume that the element being selected for is on a side proportional to the size of the side.

Assume that if we have a list of size n and split on element k , the element we are looking for has probability $(k - 1)/n$ of being on the left side, $(n - k)/n$ of being on the right side, and $1/n$ of being found. You may simplify the analysis by assuming that you split into k and $n - k$ and never find the element being selected for until $n = 1$.

- (a) Make the “back-of-the-envelope” calculation assuming that the pivot element is always exactly at $n/4$. Write a recurrence for the number of comparisons. Solve recurrence using constructive induction to get a good upper bound.
- (b) Do a more careful analysis assuming the each element is chosen as the pivot with probability $1/n$. Write a recurrence for the number of comparisons. Solve recurrence using constructive induction to get a good upper bound.
- (c) How do your results of Parts (a) and (b) compare?

Problem 3. Assume we have n distinct elements $x_1, x_2, \dots, x_n$ with non-negative weights $w_1, w_2, \dots, w_n$, such that $\sum_{i=1}^n w_i = 1$. The *weighted middle element* is the element x_k satisfying

$$\sum_{x_i < x_k} w_i < \frac{1}{2} \quad \text{and} \quad \sum_{x_i > x_k} w_i \leq \frac{1}{2}$$

- (a) Show how to compute the weighted middle element in $\Theta(n \log n)$ time using sorting.
- (b) Show how to compute the weighted middle element in $\Theta(n)$ time using a linear time selection algorithm.

Problem 4. **Challenge problem.** Show how to find the median of 5 numbers with only 6 comparisons.