Lecture 6 Scratchwork

COT 4400, Fall 2015

September 10, 2015

Identify and prove a tight upper bound on the worst-case time complexity of Selection Sort (below).

```
  Input: data: an array of integers to sort
  Input: n: the number of values in data
  Output: permutation of data such that data[1] ≤ ... ≤ data[n]
1 Algorithm: Selection Sort
2 foreach i = 1 to n do
3 |   Let m be the location of the min value in the array data[i..n]
4 |   Swap data[i] and data[m]
5 end
6 return data
```

Proof. The for loop in lines 1–4 iterates $n = O(n)$ times. Each iteration takes $O(1)$ time to perform the swap in line 3, and $O(n - i)$ time to find the max in line 2. (Our rough estimate would be that this loop takes $O(n(n - 1)) = O(n^2)$ time; however, the last few iterations are constant.)

Total runtime for this loop will be $\sum_{i=2}^{n} O(n - i) = O(n - 2) + O(n - 3) + O(n - 4) + \ldots + O(2) + O(1)$. By envelopment property, this equals $O((n - 2) + (n - 3) + \ldots + 2 + 1) = O(1 + 2 + \ldots + (n - 2)) = O(\frac{(n-2)(n-1)}{2}) = O(n^2)$.

The return statement in line 5 takes $O(1)$, so Selection Sort takes $O(n^2) + O(1) = O(n^2)$ time. $\square$

Linear recurrences with constant coefficients

Fibonacci sequence: $F(0) = 1$, $F(1) = 1$, $F(n) = F(n-1) + F(n-2)$ for all $n \geq 2$.

Characteristic polynomial: polynomial of degree k, where $T(n-k)$ is the farthest value that $T(n)$ depends on. The coefficients of the polynomial are equal to the negation of the coefficients in the recurrence equation, except for the coefficient of n^k, which is 1.

For Fibonacci, $c(x) = x^2 - x - 1$. If $T(n) = 2T(n-1) + 4T(n-2)$, $c(x) = x^2 - 2x - 4$.