CSCI 262
Data Structures

18 – Selection Sort

Sorting
- Input: a list of elements, e.g. integers
- Output: a list of the input elements in sorted order

Why do we study this problem?
- Teaching example
  - Algorithm design
  - Algorithm analysis
- Sorting is also useful for all sorts of applications

Selection Sort
- Input: a list of elements, e.g. integers
- Output: a list of the input elements in sorted order

- A simple solution:
  - Find the minimum element in the list
  - Swap it with the first element in the list
  - Sort the sublist after the first element

- This sorting algorithm is named selection sort.

Selection Sort Illustrated

Selection Sort Code

template <typename T>
void selection_sort(vector<T> & vec) {
    int n = vec.size();
    for (int left = 0; left < n; left++) {
        int right = left;
        for (int j = left + 1; j < n; j++) {
            if (vec[j] < vec[right]) right = j;
        }
        swap(vec[left], vec[right]);
    }
}

Analyzing Selection Sort
Recall we want to count basic computer steps...

1 template <typename T>
2 void selection_sort(vector<T> & vec) {
3     int n = vec.size();
4     for (int left = 0; left < n; left++) {
5         int right = left;
6         for (int j = left + 1; j < n; j++) {
7             if (vec[j] < vec[right]) right = j;
8         }
9         swap(vec[left], vec[right]);
10     }
11 }
12
What is $x$? Ans: $n - left - 1$.

How do we add these up?
Analyzing Selection Sort

Things we can easily count:
1 step (line 3)
4n steps (lines 5 and 10)

Things that are trickier:
n – left – 1 (different value of left each time)

Analyzing Selection Sort

Just have to count carefully:
1st time through:
left = 0, so n – left – 1 = n – 1
2nd time through:
left = 1, so n – left – 1 = n – 2
... Last time through:
left = n – 1, so n – left – 1 = 0

Putting it all together, we have:
Cost of selection sort is
\[ 1 + 4n + n(n - 1)/2 \]
\[ = \frac{n^2}{2} + 7n/2 + 1 \]

What is the “big-O” of this expression?

Analysis Complete

Selection sort is O(n^2)

Can we do better? (Yes, to be continued)

Up Next

- Friday, Nov. 10
  - Lab 10 - Inheritance
- Monday, Nov. 13
  - Midterm Review
  - Project 4 Due
- Wednesday, Nov. 15
  - Midterm Exam 2 (in class)
- Friday, Nov. 17
  - Fun & Games (optional)
- Monday, Nov. 20
  - Analysis of Algorithms 2 (recursive algorithms)