### CS Teaching Academy Activity 5B: Algorithm Complexity

# Why?

The appropriateness of an algorithm for a particular problem depends in part on its cost characteristics, or cost complexity. The process of choosing the best algorithm must involve this information.

## Vocabulary:

cost complexity, scalable, tractable, intractable

### Key Questions (answer individually, then check answers with your teammates):

1. How do we determine the cost complexity of an algorithm?

2. True or False? A polynomial-time algorithms, e.g.  $O(n^3)$ , is scalable.

### Exercises (answer as a team, then check with the instructor):

- 1. Why are intractable algorithms not used for real problems?
- 2. Which is most important, worst-case cost, average-case cost, or best-case cost?

### Problems (answer as a team, then write your answers on the board:

1. Sequential Search has a worst-case cost of (4 + 3n), where n is the input size. What is its average-case cost (assuming random inputs)?

```
1. int search(int[] source, key) { 1
2. for (int i=0; i<source.length; i++) { 1+n+n
3. if (key == source[i]) { n
4. return i; 1
5. }
6. return -1; 1
7. }</pre>
```

2. What would you guess the average-case cost of binary search to be, given a worst-case cost of  $(2 + 7 * \log 2(n))$ ?

3. Determine the worst-case cost of the following algorithm as a function of the problem size, n, which is equal to list.length.

```
/*
 * Recursively search for the smallest element in a list.
 */
int pl(int[] list, int start) {
    if (start < 0 || start >= list.length) {
        return -1;
    }
    if (start == list.length-1) {
        return list[list.length-1];
    }
    int n = pl(list, start+1);
    if (n < list[start]) {
        return n;
    }
    return list[start];
}</pre>
```