Lab 14: Complexity (Big O)

1. Computational Complexity of `std::sort` [10 minutes]: Code along with the GSI as we investigate the complexity of `std::sort` by timing it for various sized vectors.

   a. This exercise is interactive, and we will all be adding our times to the following spreadsheet:
   https://docs.google.com/spreadsheets/d/1qG3le5iiMsikn_jkqtPbtTpdhvDdnS2c1Fzd41LZSBU/edit?usp=sharing

2. Understanding Complexity of STL algorithms [10 minutes]: For each of the following STL algorithms, use cplusplus.com to find the complexity. You will need this on the quiz:

   a. `std::sort`

   b. `std::distance`

   c. `std::partition`

2. Determining algorithmic complexity [45 minutes]: For each code snippet, determine the computational complexity. You will be asked about the answers on the quiz. Feel free to run these examples in code to check your answer before submitting.

   A)

   ```
   int a = 0, b = 0;
   for (i = 0; i < N; i++) {
       a = a + rand();
   }
   for (j = 0; j < M; j++) {
       b = b + rand();
   }
   ```

   B)

   ```
   int a = 0;
   for (i = 0; i < N; i++) {
       for (j = N; j > i; j--) {
           a = a + i + j;
       }
   }
   ```

   C)

   ```
   int i, j, k = 0;
   for (i = n / 2; i <= n; i++) {
   ```

```
    for (j = 2; j <= n; j = j * 2) {
        k = k + n / 2;
    }
}
```

D)

```
int a = 0, i = N;
while (i > 0) {
    a += i;
    i /= 2;
}
```

E)

```
for(int i=0;i<n;i++){
i*=k;
}
```

4. Finding the nth smallest element [10 minutes]: In this problem, you will try two different methods for finding the 50,000th smallest element in a vector of size 1,000,000

   a. Using `std::sort`

   b. Using `std::nth_element`

   Write a program that compares the two. You should also check the documentation to see what their complexities are. In order to ensure the compiler doesn't optimize away your code, make sure you print the nth element. You should also be sure to fill the vector with random values, since the order of elements may affect the runtime.

Some examples borrowed from https://www.geeksforgeeks.org