CSM 61B
Fall 2020

61B Potpourri
Mentoring 13: November 16-20, 2020

1 Identify the Sort!

1.1 Each of the following sequences represent an array being sorted at some intermediate step. Match each sample with one of the sorting algorithms: **insertion sort, selection sort, heapsort, merge sort, quicksort**. The original array is below.

```
5103 9914 0608 3715 6035 2261 9797 7188 1163 4411
```

(a)
```
0608 5103 9914 3715 6035 2261 7188 9797 1163 4411
0608 3715 5103 6035 9914 2261 7188 9797 1163 4411
```

(b)
```
0608 1163 5103 3715 6035 2261 9797 7188 9914 4411
0608 1163 2261 3715 6035 5103 9797 7188 9914 4411
```

(c)
```
9797 7188 5103 4411 6035 2261 0608 3715 1163 9914
4411 3715 2261 0608 1163 5103 6035 7188 9797 9914
```

(d)
```
5103 0608 3715 2261 1163 4411 6035 9914 9797 7188
0608 2261 1163 3715 5103 4411 6035 9914 9797 7188
```

(e)
```
0608 5103 9914 3715 6035 2261 9797 7188 1163 4411
0608 2261 3715 5103 6035 9914 9797 7188 1163 4411
```

2 Berkeley Bytes

2.1 For taxes, you must to submit a list of the ages of your customers in sorted order. Define `ageSort`, which takes an **`int`**`[] array` of all customers' ages and returns a sorted array. Assume customers are less than 150 years old.

```
public class BerkeleyBytes {
    public static int[] histogram(int[] ages) {
        private static int maxAge = 149;
        int[] ageCounts = new int[maxAge + 1];
        for (int age : ages) {
            ageCounts[age] += 1;
        }
        return ageCounts;
    }
    public static int[] ageSort(int[] ages) {















    }
}
```

2.2 Time passes and your restaurant is doing well. Unfortunately, our robot overlords advanced medicine to the point where humans are now immortal.

(a) How could we extend the algorithm to accept a list of any ages?

(b) When would we be able to use this type of sort?

3 Shortest Paths

3.1 Given a weighted, directed graph G where the weights of every edge in G are all integers between 1 and 10, and a starting vertex s in G, find the distance from s to every other vertex in the graph where the distance between two vertices is defined as the weight of the shortest path connecting them, or infinity if no such path exists.

(a) Design an algorithm for solving the problem better than Dijkstra's.

(b) Give the runtime of your algorithm.

4 Largest Perimeter

4.1 Given an array A of positive lengths, return the largest perimeter of a triangle with non-zero area, formed from 3 of these lengths. Recall the Triangle Inequality, which states that for any triangle, the sum of the lengths of any two sides must be greater than or equal to the length of the remaining side ($a + b > c$). If it is impossible to form any triangle of non-zero area, return 0.

For example, A = [2, 1, 2] returns 5. A = [1, 2, 1] returns 0. A = [3, 2, 3, 4] returns 10.

What is the runtime of your solution?

```
public int largestPerimeter(int[] A) {











}
```

Note: this problem was adapted from LeetCode (https://leetcode.com/problems/largest-perimeter-triangle/).