1   Warmup

Algorithm	Best-case	Worst-case	Stable
Selection Sort			
Insertion Sort			
Merge Sort			
Quicksort			
Heapsort			

1.1   Give a best and worst case input for insertion sort.

1.2   Do you expect selection or insertion sort to run more quickly on a reverse list?

1.3   Give a worst case input for quicksort. Assume that we're always picking the leftmost item as our pivot.

1.4   Why does Java's built-in `Array.sort` method use quicksort for `int`, `long`, `char`, or other primitive arrays, but merge sort for all `Object` arrays?

2 Inversions

We have a list of N elements that should be sorted, but to our surprise we recently discovered that there are at most k pairs out of order, or k **inversions**, in the list. As a small example, the list { 0, 1, 2, 6, 4, 5, 3 } contains 5 inversions: $(6,4), (6,5), (6,3), (4,3), (5,3)$.

3 Decisions

3.1 Given a list of objects that is totally sorted except for $K < \log N$ randomly chosen elements that are out of place, choose the best sorting algorithm to solve the problem and give a tight asymptotic runtime bound.

3.2 What's the best sorting algorithm to apply on an array of N randomly distributed objects? Give a tight asymptotic runtime bound as well.

4 What's that sort?

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

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

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