1. Describe what is meant by "divide and conquer".

2. The following function incorrectly merges two lists, what's wrong?

```
def merge(sorted1, sorted2):
      result = []
      index1 = 0
      index2 = 0
      while index1 < len(sorted1) and index2 < len(sorted2)
            result.append(sorted1[index1])
            index1 = index1 + 1
            result.append(sorted2[index2])
            index2 = index2 + 1
      if index1 < len(sorted1)
            result.extend(sorted1[index2:])
      otherwise if index2 < len(sorted2)
            result.extend(sorted2[index1:])
      return result
```

3. Draw a tree that shows schematically how the data is organized when performing a merge sort on the following data:

```
[2, 6, 8, 7, 9, 5, 0, 1, 3, 4]
```

Assume the splits happen using the even/odd index approach used in lecture.

4. What is the best, worst, and average time complexity for merge sort?

5. Given the partition function for quick sort, trace what the return result will be if `testPartition` is passed `L=[2, 5, 1, 3, 4, 0]`.

```
def partition(pivot, L)
      (less, same, more) = ([], [], [])
      for e in L
            if e < pivot
                  less.append(e)
            otherwise if e > pivot
                  more.append(e)
            otherwise
                  same.append(e)
      return (less, same, more)
```

```
def testPartition(L):
      pivot = L[0]
      (less, same, more) = partition(pivot, L)
      return less + same + more
```

6. Show the output for `quickSort` for the list `[2, 5, 1, 3, 4, 0]`:

```
def quickSort(L):
    if L == []:
        return []
    else:
        pivot = L[0]
        less, same, more = partition(pivot, L)
        print (less, same, more)
        return quickSort(less) + same + quickSort(more)
```

7. Explain the worst case scenario for quicksort using pivot first. Give an example list of 6 elements that demonstrates this.

8. Recall the sorting lab from 141. You used an algorithm called k-select, that selected the median value from an unsorted list. How is k-select similar/related to quicksort.