Joe Skimmons, jws2191

1a.

```
int sum = 0;
for ( int i = 0; i < n ; i ++)
   for ( int k = i ; k < n ; k ++)
      sum = sum + 1;
```

Big Oh Runtime: : O(n^2)

1b.

```
int sum = 0;
for ( int i = 0; i < 23; i ++)
   for ( int j = 0; j < n ; j ++)
      sum = sum + 1;
```

Big Oh Runtime: O(n)

1c.

```
public int foo(int n, int k) {
   if(n<=k)
      return 1;
   else
      return foo(n/k,k) + 1;
}
```

Big Oh Runtime: O(log(n))

2. Write Java-like pseudocode (or actual Java code) for the method List union(List l1, List l2) that computes l1 ∪ l2, i.e. the union of l1 and l2, using only basic list operations. The resulting list should not contain any duplicates. You do not have to provide a surrounding class.

```
1  public List union(List l1, List l2){
2      notContained = True;
3      List union = new List();
4      // Adds all of l1 to the new list
5      for (int x=0;x<l1.size();x++) {
6          union.add(l1.get(x));
7      }
8      // only adds from l2 to the union list if it is not already contained
9      for (int x=0;x<l2.size();x++) {
10         // compares current value to every value in union, to see if it is contained
11         // does this for every value in l2
12         for (int y=0;y<union.size;y++) {
13             if (l2.get(x).equals(union.get(y))){
14                 notContained = false;
15             }
16         }
17         // adds it only if it is not in union alreadt
18         if(notContained){
19             union.add(l2.get(x));
20         }
21         notContained = True;
22     }
23     return union;
24 }
```

3a.

1. Move 4 from the input track to the first holding track
2. Move 3 from the input track to the second holding track
3. Move 1 from the input track to the third holding track
4. Move 8 from the input track to the first holding track
5. Move 6 from the input track to the second holding track
6. Move 2 from the input track to the third holding track
7. Move 7 from the input track to the third holding track
8. Move 9 from the input track to the output track
9. Leave 5 on the input track
10. Move 8 from the first holding track to the output track
11. Move 7 from the third holding track to the output track
12. Move 6 from the second holding track to the output track
13. Move 5 from the input track to the output track
14. Move 4 from the first holding track to the output track
15. Move 3 from the second holding track to the output track
16. Move 2 from the third holding track to the output track
17. Move 1 from the third holding track to the output track

3b. A train of 941235678 is a series that could not be solved with three shunting yards because there is no way to arrange the cars on the tracks from least to greatest, when placed in the shunting yards there must be greater numbers on the bottom, making them impossible to reach and the sequence unsolvable.