CS 1713 Exam 3 Solutions — Spring 2010

1a) (5 points)

```
public int linearSearch(String[] array, String target) {
   for (int i=0;i<array.length;i++)
      if (array[i].equals(target))
         return i;
   return -1;
}
```

1b) (15 points)

```
public int binarySearch(String[] array, String target) {
   int low = 0;
   int high = array.length - 1;
   while (low <= high) {
      int mid = (low + high)/2;
      if (array[mid].equals(target))
         return mid;
      else if (array[mid]compareTo(target) < 0)
         low = mid + 1;
      else
         high = mid - 1;
   }
   return -1;
}
```

2) (20 points)

```
   public static void insertionSort (double [] list) {
      for (int index = 1; index < list.length; index++)
         insertItem(list, index);
   }

   public static void insertItem(double [] list, int index) {
      double key = list[index];
      int position = index;

      while ((position > 0) && (key < list[position-1]))   {
          list[position] = list[position-1];
          position--;
       }
       list[position] = key;
   }
```

3) (10 points) a) 18 seconds b) 57.6 seconds c) 9.6 seconds

4a) (3 points)

```
public int firstPos(String s1, String s2) {
  return s1.indexOf(s2);
}
```

4b) (7 points)

```
public String afterString(String s1, String s2) {
  int pos = s1.indexOf(s2);
  if (pos == -1)
     return "";
  return s.substring(pos + s2.length());
}
```

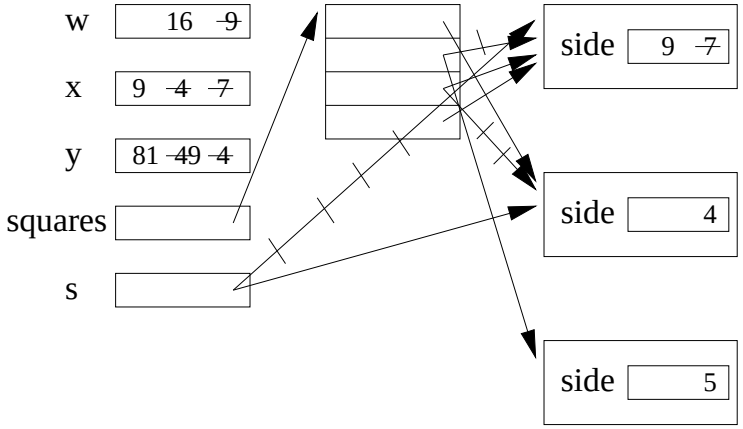
5) (10 points)

```
public void replaceLargeWithReciprocal(RationalNumber[] list) {
   for (int i=0;i<list.length;i++)
      if (list[i].getNumerator() > list[i].getDenominator())
         list[i] = new RationalNumber(list[i].getDenominator(),
                                      list[i].getNumberator);
}
```

6) (10 points)

```
public boolean checkWinnerRow(char[][] board) {
   return checkWinnerRow(board,'X') || checkWinnerRow(board,'Y');
}
public boolean checkWinnerRow(char[][] board, char c) {
   for (int i=0;i<board.length;i++)
      if (checkWinnerRow(board[i],c)
          return true;
   return false;
}
public boolean checkWinnerRow(char[] row, char c) {
   if (row.length == 0)
      return false;
   for (int i=0; i<row.length;i++)
      if (row[i] != c)
         return false;
   return true;
}
```

7) (10 points)



8) (12 points)

Selection Sort

0	1	2	3	4	5	6	7
75	**62**	**29**	**19**	**48**	**91**	**18**	**45**
18	62	29	19	48	91	75	45
18	19	29	62	48	91	75	45
18	19	29	62	48	91	75	45
18	19	29	45	48	91	75	62
18	19	29	45	48	91	75	62
18	19	29	45	48	62	75	91
18	19	29	45	48	62	75	91

Insertion Sort

0	1	2	3	4	5	6	7
75	**62**	**29**	**19**	**48**	**91**	**18**	**45**
62	75	29	19	48	91	18	45
29	62	75	19	48	91	18	45
19	29	62	75	48	91	18	45
19	29	48	62	75	91	18	45
19	29	48	62	75	91	18	45
18	19	29	48	62	75	91	45
18	19	29	45	48	62	75	91