

Problem Set 3

Specification

Submit a java file named `YourNamePset3.java`. Your file should **NOT** execute any code. You are just filling them in. When I run your program there should be no errors and no output. You may choose to test your methods as you go inside of a main method, but be sure to delete the main method before submitting. Comment your problems with the problem number above the method header. Do not use built in methods that are not in the Java Quick Reference guide.

Problem 1

Write a method that searches through an unsorted integer array, *arr*, for a *key* value. Return the index of *key* at the first index at which it is present or return -1 if *key* is not found.

```
linSearch(new int[] {1, 3, 1, 2, 4}, 1); // 0
linSearch(new int[] {1, 3, 1, 2, 4}, 3); // 1
linSearch(new int[] {1, 3, 1, 2, 4}, 5); // -1
```

Problem 2

Write a method that binary searches through a sorted `String` array, *arr*, for a *key* value. Return the index of the first instance of *key* or return -1 if *key* is not found.

```
binSearch(new String[] {"apple", "banana", "lime", "yucca"}, "lime"); // 2
binSearch(new String[] {"apple", "banana", "lime", "yucca"}, "lemon"); // -1
```

Problem 3

Given an unsorted integer array, *arr*, sort the array in-place using a selection sort.

```
int[] arr = {2, 3, 1, -1, -4};
selSort(arr) // sorts to [-4, -1, 1, 2, 3]
```

Problem 4

Given an unsorted integer array, *arr*, sort the array in-place using a bubble sort.

```
int[] arr = {2, 3, 1, -1, -4};
bubbleSort(arr) // sorts to [-4, -1, 1, 2, 3]
```

Problem 5

Given an unsorted integer array, *arr*, sort the array in-place using an insertion sort.

```
int[] arr = {2, 3, 1, -1, -4};
insSort(arr) // sorts to [-4, -1, 1, 2, 3]
```

Problem 6

Given an unsorted integer array, *arr*, sort the array in-place using a shell sort. You can use shells of size $3x + 1$, 1, 4, 13, 40, 121, 364, 1093,

```
int[] arr = {2, 3, 1, -1, -4};
shellSort(arr) // sorts to [-4, -1, 1, 2, 3]
```

Problem 7

Given an **int** array, *arr*, write a method that returns **true** if the values are sorted least to greatest or greatest to least and **false** otherwise.

```
isSorted(new int[] {1,2,3}); // true
isSorted(new int[] {3,2,1}); // true
isSorted(new int[] {1,3,2}); // false
```

Problem 8

You get a job at the Olympics ensuring that the gymnastics event is scored correctly. A gymnast's score is determined by a panel of 6 judges who each decide a score between 0.0 and 10.0. The final score is determined by discarding the high and low scores, and averaging the remaining 4. Write a method that returns the correct score.

```
gymnasticScore(new double[] {5, 5.5, 10.0, 9.0, 3.0, 6.5}); // 6.5
```

Problem 9

A professional gambler living in Las Vegas keeps track of their gambling winnings each day. Each day they store their winnings inside of an array. They want to know their hottest streak and come to you for help. A streak will be defined as contiguous elements within the array, i.e. a subarray where the elements are successive. Assume the array is written in units of 100 dollars. Write a method that takes an array *arr* and returns the amount won during the hottest streak.

```
hottestStreak(new int[] {-1, 2, -4, 2, -1, 2, 5, -5}); // 800
// note 800 is returned because each element represents $100
// the hottest streak was 2,-1,2,5 = 8
```

Problem 10

Given an integer array sorted in non-decreasing order, *arr*, there is exactly one integer in the array that occurs more than 25 percent of the time. Return that integer.

```
moreThanQuarter(new int[] {1, 2, 2, 6, 6, 6, 6, 7, 10}); // 6
moreThanQuarter(new int[] {1, 1});                      // 1
```

Problem 11

Given a fixed-length integer array, *arr*, duplicate each occurrence of zero, shifting the remaining elements to the right. Note that elements beyond the length of the original array are not written. Do the above modifications to the input array in-place and do not return anything.

```
shiftElements(new int[] {1, 0, 2, 3, 0, 4, 5, 0}); // [1, 0, 0, 2, 3, 0, 0, 4]
shiftElements(new int[] {1, 2, 3});               // [1, 2, 3]
```

Problem 12

Given an array, *arr*, with *n* objects colored green, white, or red, sort them so that objects of the same color are adjacent, with the colors in the order green, white, and red.

We will use the integers 0, 1, and 2 to represent the color white, red, and green, respectively. The algorithm should needs to sort the array in-place and scale as $\mathcal{O}(n)$.

```
sortColors(new int[] {2, 0, 2, 1, 1, 0, 1}) // [2, 2, 0, 0, 1, 1, 1]
```

Problem 13

The `indexOf` method for strings returns the index of the first occurrence of a substring *sub* within a string *str*. If the substring is not found, the method returns -1. Write a method that uses a linear search to perform the same functionality.

```
indexOf("hello", "ll"); // 2
indexOf("hello", "x");  // -1
```

Problem 14

Suppose an array, *arr*, of length *n* is sorted in ascending order and each element is rotated right between 1 and *n* times. For example, the array *nums* = [0, 1, 2, 4, 5, 6, 7] might become [4, 5, 6, 7, 0, 1, 2] if rotated 4 times, and [0, 1, 2, 3, 4, 5, 6, 7] if rotated 7 times. Given that *arr* contains unique elements, return the minimum element of the array. Your algorithm must scale as $\mathcal{O}(\log_2(n))$.

```
rotatedMin(new int[] {3, 4, 5, 1, 2}); // 1
rotatedMin(new int[] {4, 5, 6, 7, 3}); // 3
```

Problem 15

You're at Cedar Point and are in line for the Millenium Force rollercoaster. Each person wears a sticker indicating their initial position in the queue from 1 to *n*. Any person can bribe the person directly in front of them to swap positions, but they still wear their original sticker. One person can bribe at most two others.

Determine the minimum number of bribes that took place to get to a given queue order. Return the number of bribes, or if anyone has bribed more than two people, return -1 .

```
minBribes(new int[] {1, 2, 3, 5, 4, 6, 7, 8}) // 1
minBribes(new int[] {4, 1, 2, 3})           // -1
minBribes(new int[] {2, 1, 5, 3, 4})         // 3
```

Problem 16

There are *n* poms arranged on a rope. There is a String array *colors* that represents the color of the pom at position *i*. You want to make sure that no two consecutive poms are the same color, so you remove poms until you get a unique array of poms. Unfortunately, each pom requires a different amount of time to remove. An array of integers *neededTime* holds the amount of time required to remove the *i*-th pom. Given *colors* and *neededTime*, return the minimum amount of time required to ensure that the rope has no poms of the same color touching.

```
removePoms(new String[] {"red","blue","red","red","green"}, new int[] {1,2,3,4,5})
// 3
removePoms(new String[] {"red","red","blue","red","red"}, new int[] {2,3,4,1})
// 3
removePoms(new String[] {"red","red","blue","blue","green","red","red"}, new int[]
{1,2,3,4, 5,6,7}) // 10
```

Problem 17

Dr. B is working on his latest piece of abstract art: a mural consisting of an array of crescent moons and fishing hooks. Unfortunately, the greedy English department teachers are claiming that moons look like an uppercase C and fishing hooks look like a J, and they have a copyright on CJ and JC. Therefore, for each time CJ appears in the mural, I have to pay *fine1* dollars, and for each time JC appears in the mural, I must pay *fine2* dollars. Assume that I have a mural painted partially, represented by a string of C's, J's, and question marks. I want to

minimize the cost I have to pay, so write a method that tells me what the minimum I need to pay is, given the fines and the current state of the mural.

```
fineCalculator(fine1, fine2, mural);  
fineCalculator(2, 3, "CJ?CC?"); // 5  
fineCalculator(2, 5, "??J??"); // 0  
fineCalculator(1, 3, "C?J"); // 1
```

Problem 18

Mrs. K is having a contest where students can guess the number of orders at River Roasters for a particular month. The student that won last month is awaiting her check in the mail. Unfortunately, the RRHS print shop computer is broken and the 6 key is broken. To ensure that we can print comically large checks for the winning student, we decide to pay her the N dollar grand prize with two checks whose value adds up exactly to the total prize. The two check amounts will not use the number 6 anywhere. Given a grand prize number N print the two check values that should be printed so that the winner receives the total prize.

```
prizeMoney(363); // 333 30  
prizeMoney(6666); // 3333 3333  
prizeMoney(7636); // 7333 303
```

Problem 19

The Empire's new Death Star works according to a series of computational instructions. The laser can either shoot, or charge. The laser starts with a power of 1 and each time the laser is charged, its power is doubled. The Death Star's instructions are stored in an array as strings of S and C, for shoot and charge. The instruction sequence ["S", "C", "C", "S", "S"] will do 1 damage, charge to 2 damage, charge to 4 damage, then shoot for 4 damage twice. The effects of charging persist after shooting.

The Jedi find a way to protect their base by using the force. The protective shield can protect against exactly D damage ($D + 1$ will break the shield). But this may not be enough! So one Jedi gets access to the array and can swap two adjacent instructions without anyone noticing. Given a string array of instructions called *instructions* and int value of D , return the minimum number of swaps required to ensure that the shield holds. If impossible, return -1

```
swapsToSave(new String[] {"S", "C", "C", "S", "S", "C"}, 6); // 2  
swapsToSave(new String[] {"S", "S"}, 1); // -1  
swapsToSave(new String[] {"C", "S", "C", "S", "S"}, 3); // 5
```

Problem 20

The rebels have intercepted sith communications. The issue with their electronics is that the information appears in a random order. The rebel computer scientists determined that each message is split into several lines where each line contains a message number, followed by a line number, and then finally a part of the

message. It is up to you to write a program that takes a text file and restores the message. The text file given to you has the following format: The first line of input consists of the number of lines captured. Each line has three components: an integer representing the message number this line belongs to, a second integer representing the line number within the message, and the final component is a string representing the message component itself. Your program should output a message, in order. Hint: Make a `Message` object with message number and line number attributes then read the data into a `Message[]` array. Write a method to sort the message array according to the rules using one of the sorts we learned in class.

A text file containing the following as input:

```
5
0 1 the shadows. This ultimately led to Anakin's fall as a Sith
1 0 After seeing her in action, Han offered Rey a permanent
0 0 Unbeknownst to Anakin, the Emperor was manipulating him from
1 1 position as a member of his crew.
0 2 lord.
```

Produces the following output:

```
Message 0:
Unbeknownst to Anakin, the Emperor was manipulating him from
the shadows. This ultimately led to Anakin's fall as a Sith
lord.
Message 1:
After seeing her in action, Han offered Rey a permanent
position as a member of his crew.
```