Qiman Wang

Q1.

I love clothing! Almost all millennials love clothing and wants to be fashionable thanks to social media and online shopping. My closet is filled with clothing and I want to get rid of some clothing and make some money.

A problem I have and many have is too much clothing. My closet is full, my basement has a lot of my old clothe. I don’t want to wear it because I don’t like it anymore. But I don’t want to throw it away.

We can create a online auction system for clothing. People can post their old clothing and other’s might really like something that I don’t like anymore. It’s a win-win situation. The auction system will allow users to post their clothing pictures., our algorithms will automatically determine your recommended buyers and recommend it, also price at reasonable price.

The site is easy to create. We need about 5 software engineer and takes about 6 month to deploy on app store. So the cost is rent (24,000) + salary (300,000) + equipment (20,000). We are looking at about $350,000.

If this works out well, we can create a auction platform for other things too like video games, books, etc. If it doesn’t work out well, we can create a generic auction site like eBay but make it better, somehow.

Q6c.

//Warning: Binary search, takes in a sorted array in non-decreasing order, and a target value.

public static int binarySearchTree(int[] arr, int targetValue) {  
 int leftIndex **=** 0; // left most index  
 int rightIndex **=** arr.length **-** 1; //right most index  
 while (leftIndex **<=** rightIndex) {  
 int idxDifference **=** rightIndex **-** leftIndex;  
 int midIndex **=** leftIndex **+** (idxDifference **/** 2);  
 if (targetValue **<** arr[midIndex]) {  
 rightIndex **=** midIndex **-**1; // eleminate midIndex element and  
 //all element to right of midIndex  
 }  
 else if(targetValue **>** arr[midIndex]){  
 leftIndex **=** midIndex;//eleminate all element to the left of midIndex  
 //and midIndex element  
 }  
 else return midIndex; // found the index of element in array  
  
 }//while loop keep running as long as leftIndex not overlap rightIndex.  
 return **-**1; //if not found element in array  
 }//binary search method

\*\*incase the code above is hard to read, I added a dark background.

public static int binarySearchTree(int[] arr, int targetValue) {  
 int leftIndex **=** 0; // left most index  
 int rightIndex **=** arr.length **-** 1; //right most index  
 while (leftIndex **<=** rightIndex) {  
 int idxDifference **=** rightIndex **-** leftIndex;  
 int midIndex **=** leftIndex **+** (idxDifference **/** 2);  
 if (targetValue **<** arr[midIndex]) {  
 rightIndex **=** midIndex **-**1; // eleminate midIndex element and  
 //all element to right of midIndex  
 }  
 else if(targetValue **>** arr[midIndex]){  
 leftIndex **=** midIndex;//eleminate all element to the left of midIndex  
 //and midIndex element  
 }  
 else return midIndex; // found the index of element in array  
  
 }//while loop keep running as long as leftIndex not overlap rightIndex.  
 return **-**1; //if not found element in array  
 }//binary search method