Assorted Questions and Concepts – Part 2

Cpt S 321

Washington State University

Question 1

- You have an array of ushort numbers and you need to count the number of distinct values. Implement the function below such that:
    - It runs in O(n) time complexity
    - It runs with O(1) storage complexity

```
static int CountDistinct(ushort[] numbers)
{

}
```

Question 1 Concepts

- Need to recognize that this requires some kind of table to keep track of values that you’ve seen
    - A hash set would work fine, but with the limited range of ushort values we don’t need one.
    - What can we use instead?

Question 1 Concepts

- Boolean table works just fine
- 65,536 possible unique ushort values (=2^{16})
- We can just use ushort.MaxValue + 1 for allocation size
- Go through all the numbers in the array and mark them as used in a boolean table
- Then count the number of used items in the table

Question 2

- Implement a function that takes a ushort[] array and returns a sorted list of values representing all values from the array that repeated (occurred at least 2 times).
- Example: If the array was: { 1, 2, 9, 8, 7, 7, 8, 5, 9, 1, 3, 3, 3, 9, 1 }
- Then the returned list would contain (in this order): {1, 3, 7, 8, 9}

```
/// <summary>
/// Makes a sorted list of values from nums that occurred at least 2 times.
/// </summary>
static IList<ushort> GetRepeatsSorted(ushort[] numbers)
{

}
```

Question 2 Requirements

- Implement with worst case O(n) time complexity
  - You may assume that if you use the System.Generics.List class that the Add function always runs in constant time (even though in many cases that’s not true)
- Implement with worst case O(n) space complexity
  - Should be obvious why

Question 2 Concepts

- Need to know how to build a frequency table. This is a table that contains a count of # of occurrences for each value in the data set.
- With ushort values being used again, our frequency table can be allocated with O(1) space complexity
- If we were dealing with an array of strings we'd probably want to build the frequency table with a Dictionary.
  - var frequencyTable = new Dictionary<string, int>();
- But in this case we're simply dealing with ushorts

Question 4

- Count the number of distinct values in an array of unsigned integers.
- Can’t do this in the same way we did it on the array of ushorts.
- This would get marked as wrong:
  - int[] frequencies = new int[(long)uint.MaxValue + 1];
- Can’t just assume infinite memory
- How much memory are we allocating for that frequency table?

Question 4

- 2^32 slots in frequency table and 4 bytes per integer, means the table is:
    - 2^32*4 = 17179869184 bytes
    - = 16777216 kb
    - = 16384 mb
    - = 16 gigabytes
- Operating system will deny that allocation request (probably even on a machine that has over 16 GB of ram)

Summary

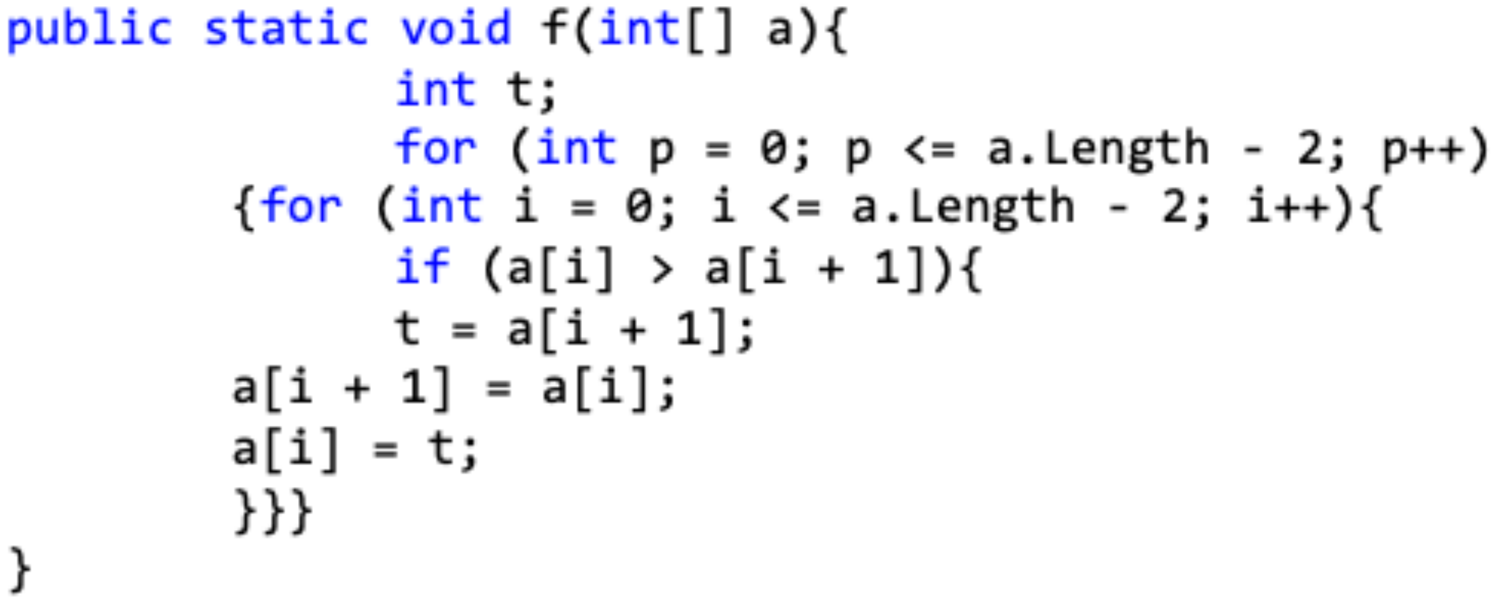
- EVERYTHING within these slides should make good sense to you
- These are the types of things that you'll need to know how to do for homework assignments, exams, and many practical software engineering problems
- These are just a small handful of examples. We will have more questions throughout the semester and we'll look at several that are much more difficult.

What to expect in Mid-term 1

- Multiple choice questions
- Read, understand, and describe the output of a given program
- Implement an algorithm given complexity constraints
- Design a solution given a problem
- Writing test (the smart way!)
- Comment on the quality of code
- Fix wrong code

Examples of questions

- You are maintaining a software and you come across the following method. Comment on its quality by pointing out problems that you see and rewrite the method by fixing them. (Note: Feel free to do the changes in multiple iterations if you find it easier.)



```
public static void f(int[] a){
              int t;
              for (int p = 0; p <= a.Length - 2; p++)
        {for (int i = 0; i <= a.Length - 2; i++){
              if (a[i] > a[i + 1]){
              t = a[i + 1];
        a[i + 1] = a[i];
        a[i] = t;
        }}}
}
```

Examples of questions (cont.)

- Consider the following implementation of method ____ and the corresponding test cases that were written by a developer based on the specifications of the method. How confident are you that the method does what it is supposed to do? Please justify your answer. If you think that additional tests must be added please write those and justify the necessity for each one of them. (Note: No need to change the implementation of the method.)

Examples of questions (cont.)

- Using TDD, implement a method that takes as input a list of unsigned integers (uint) and returns the number of district integers in that list. Do not alter the list. Explicitly state the number of commits you are having, the content of each commit, and the commit message.

Example of “design and implement” problem

You are contacted by a company to build an application that will allow users to create shapes based on an input sequence of characters and then manipulate them. As a start, the application should support three shapes: circle, square, and rectangles. More will definitely come later so your design should allow easily adding those.

For example, the sequence “c s c r” corresponds to the user wanting to create a circle, a square, a circle, and a rectangle. The first shape will always be created with a default size (of your choice but can be changed by the user). The next will be created by doubling the default size. The third shape – by tripling the default size, etc. The sequence of shapes entered by the used should be stored in an appropriate data structure.

The user can manipulate the shapes in different ways. For instance, he should be able to Compute the area of a sequence of shapes, which is the cumulative area of all shapes.

Filter shapes on a specific criteria, such as the area of the shape. For example, the user might want to know what are the shapes whose area is less than or equal to a certain threshold.