Final Written HW

CIS 121—Fall 2017

Due: Wednesday, December 6, 2017

This is a closed book exam. No computers or internet-connected devices are allowed during the exam. You are permitted to use one 8.5”x11” page with handwritten notes (these can be on both sides of the paper). Please write your name on your page of notes and turn it in at the end of the exam. If you need scratch paper, please get some from the front of the classroom. There are 10 questions on this exam, plus 4 extra credit questions, for a total of 14 questions.

Your name:

Your PennKey (i.e. ccb):
1 **CIS 121 made you a better programmer**

45 points

Analyze each of the snippets of Java code below. For each of the snippets, please answer the 3 questions about it:

1. What is the order of the growth in terms of Big O (with respect to the length of the list or the Strings) for the code as it is formulated? In addition to giving the Big O, please provide a short justification saying how you arrived at your answer (1 or 2 sentences).

2. How could it be improved using a data structure or algorithm from class? Say which data structure or algorithm you would use to improve it. Also give a short description of how you would use it to solve the problem. Just give your answer in plain English; do not write Java code or pseudo code.

3. Say how your change would improve the runtime. In addition to giving the Big O, please provide a short justification saying how you arrived at that analysis (1 or 2 sentences).

**Java code snippet 1:** The goal of this is to check whether an item is in a list. Here, the list of items is already sorted when passed into the function. In your analysis say whether `assert(isSorted(items));` should count towards the run time or not. If so, how much time does it take?

```java
public <V extends Comparable<V>> boolean containsItem(V item, V[] items) {
    assert(isSorted(items));
    int N = items.length;
    for (int i = 0; i < N; i++) {
        if (items[i].equals(item)) {
            return true;
        }
    }
    return false;
}
```

Your answers to the 3 questions:

1. 

2. 

3.
**Java code snippet 2:** The goal of this is to reverse a string, so that “final exam” would become “maxe lanif”.

```java
public String reverseString(String str) {
    String rev = "";
    for (int i = str.length() - 1; i >= 0; i--) {
        rev += str.charAt(i);
    }
    return rev;
}
```

Your answers to the 3 questions:

1. 

2. 

3. 

**Java code snippet 3:** The goal of this is take a list of all the children in Philadelphia and print out the ones that are on Santa’s nice list.

```java
public void intersectLists(Kid[] phillyKids, Kid[] santasNiceList) {
    for (int i = 0; i < phillyKids.length; i++) {
        Kid phillyKid = phillyKids[i];
        for (int j = 0; j < santasNiceList.length; j++) {
            Kid goodKid = santasNiceList[j];
            if (phillyKid.equals(goodKid)) {
                System.out.println(phillyKid);
            }
        }
    }
}
```

Your answers to the 3 questions:

1. 

2. 

3.
**Java code snippet 4:** Return the top k scores for on the final exam.

```java
public class HighestFirst implements Comparator<Exam>{
    // details omitted
}

public Exam[] getTopScores(Exam[] exams, int k) {
    java.util.Arrays.sort(exams, new HighestFirst());
    Exam[] topScoringExams = new Exam[k];
    for(int i = 0; i < k; i++) {
        topScoringExams[i] = exams[i];
    }
    return topScoringExams;
}
```

Your answers to the 3 questions:

1. 

2. 

3. 
Java code snippet 5: Auto-complete. Google shows the auto-complete of the most common queries after users have typed in a few words or letters that are a prefix match of a common query. The `startsWith` is a built-in method in Java’s String class. You should state your assumptions about its running time.

```java
public String[] autocomplete(String prefix, String[] commonQueries, int maxMatches) {
    String[] matches = new String[maxMatches];
    int counter = 0;
    int N = commonQueries.length;
    int k = prefix.length;
    for (int i = 0; i < N; i++) {
        String query = commonQueries[i];
        if (query.startsWith(prefix)) {
            matches[counter++] = query;
            if (counter == maxMatches) {
                return matches;
            }
        }
    }
    return matches;
}
```

Your answers to the 3 questions:

1. 

2. 

3. 
2  Asymptotic analysis of algorithms
   16 points

For each of the following statements, say whether it is true or false. Give a one sentence explanation of why that it so.

1. To establish a worst case analysis for an algorithm, we should analyze its performance on a single, randomly drawn input to get a representative example of the possible inputs.

2. Implementing a specific algorithm is a common way of establishing a lower bounds for a problem.

3. If a problem that is $\Omega(N^3)$ has an algorithm that is $O(N^3)$, then the algorithm is optimal.

4. The $O$, $\Omega$ and $\Theta$ notations all ignore lower order terms but keeps constants.

5. If an algorithm is $O(N^2)$ that means its average running time is quadratic with respect to the input.

6. Given an algorithm implementation, one can always derive a big-Theta bound for its runtime.

7. Amortized runtime analysis is the same as average runtime analysis

8. A $O(N^2)$ algorithm could run faster/perform fewer operations than a $O(N)$ algorithm for same sized inputs.

3  Huffman coding
   10 points

Draw a trie representing the Huffman code for the letters with the following frequencies, and give the encoding for each letter.

<table>
<thead>
<tr>
<th>Letter</th>
<th>Freq</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

4  Compression
   24 points

Give a short answer to each of these questions, along with a 1 sentence explanation of your answer.

1. Does a balanced binary tree correspond to an optimal prefix-free code?

2. How do we measure the effectiveness of a compression algorithm?

3. What key part of Huffman allows it to produce an encoding that is shorter than the input?

4. What key part of LZW allows it to produce an encoding that is shorter than the input?

5. Do all compression methods require that we have to transmit a table containing its codewords like we do with Huffman coding?

6. What can cause ambiguity to arise when we’re decoding a binary stream?

7. What strategies can we use to avoid that ambiguity?
5  Dijkstra’s algorithm for shortest paths  
   22 points

Apply Dijkstra’s algorithm to the graph above, starting at vertex 0.

1. Describe how Dijkstra’s algorithm works.

2. What is edge relaxation?

3. Fill out a table that shows the distance to each vertex, and the edge that was taken to get there.

4. Draw the shortest paths tree and label the distance from 0 to each vertex.

5. Which vertex had the most updates to its distance through edge relaxation?

6. When do we know we’ve found the shortest path to vertex x?

6  Make money with an app  
   20 points

An entrepreneurial team of Penn undergrads builds an app using Dijkstra’s algorithm that lets drivers find the shortest distance to their destination.

1. How is the system of roads represented in the graph? What are the vertices and edges? What type of graph is it?

The US congress has decided that in order to reduce the national deficit, all roads are now toll roads. Users of the app demand an update that will let them find the lowest cost route to their destination in terms of dollars instead of miles.

2. What do the Penn students need to do to update their graph?

President Trump decides to name the best, classiest road in each American city after himself. He decides that instead of charging drivers a toll, the government will pay drivers whenever they drive on any street named Trump Avenue.

3. Assuming that the graph automatically updates each time the tolls change, what happens to the directions provided by app? Will drivers still find the lowest cost ($) route to their destination?

4. How could the Penn students determine if the app could be fixed, or if it is broken beyond all repair? If it is fixable, what is the solution?

5. If the app can no longer provide driving directions, how could the Penn students still use it to make money for themselves? What would the implications be for the US economy?
Construct a deterministic finite-state automaton (DFA) for the pattern “GOOGOL”. Fill in the state transition table, draw the DFA (labeling each transition with its letter), and answer the questions below.

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. When the DFA is state \( j \) and it encounters a match character, what state does it move to?

2. When the DFA is in state \( j \) and it encounters a mismatch character, how do you determine what state it should move into?

3. Our table only shows the transitions for the 3 unique letters in our pattern. What state we should transition to if we encountered a different letter in the text, like “E”?

4. What is the name of the algorithm that we studied to construct a DFA from a pattern, and what is its run-time?
8  Preserving national security and privacy

10 points

The NSA and the FISA court which oversees it are struggling to come to an agreement on a way to preserve privacy rights while upholding national security. At issue is the fact NSA would like to monitor all messages send by email and text anyone in the world in order to detect a key phrase like “sleeper cell attack” that might indicate that an attack is imminent (in which case they would raise the terror threat level).

1. You are the judge in charge advocating for the rights of citizens, and deciding what the NSA should be allowed to do. Assuming that the NSA has the capability of monitoring all transmissions as they stream by, make an argument the NSA should be allowed to search for “sleeper cell attack” using a DFA, and that the use of a DFA is sufficient to ensure the right of citizens to keep their communications private.

2. You are a defense contractor working for the NSA. A judge has granted the NSA the right to search the stream for any number of key phrases, and to raise the terror threat level, provided they use a finite state machine to do so. If the NSA has a list of high-risk cities, and it wants to detect a set of patterns like “attack [U.S. CITY NAME] [tomorrow or next week or next month]”, would a DFA be effective? If so, how? If not, what type of machine would you construct?

9  Matching parentheses

10 points

Design an algorithm to say whether a string contains has all of its parentheses properly balanced or not. Here are several example inputs, and the value that your algorithm should return:

- ()()() - true
- { (the [small] cat) } - true
- (the cat - false
- () - false

10  Trivia

3 points

1. Who said “Object-oriented programming is an exceptionally bad idea that could only have originated in California”?

2. Which of these data structures were invented by one of the textbook authors? R-way tries, TSTs, hashing, 3-way quicksort.

3. Which of the following awards has your professor won? The MacArthur Award, the Sloan Foundation Fellowship, the Turing Award.

11  Seam Carving

15 points

Computer science researchers came up with a very cool algorithm that they published at SIGGRAPH that allows them to dynamically resize an image, so that it can display on cell phones and web browsers. The
method tries to resize an image with minimal distortion by removing one pixel from each row in the image (to narrow its width). Instead of simply selecting one row to cut, the algorithm can pick any pixel to remove. The researchers were able to automatically assign a value to each pixel that represents how perceptible it would be if removed, called its ‘energy function’. To narrow an image, they find a seam. The seam is a path of pixels, each connected to one of its eight neighbors, that goes from the top of the image to the bottom of it. The pixels in the seam are removed to narrow an image.

Given the energy function for every pixel in the image, how would you find the best seam to carve out of the image? Please include the following information in your answer:

1. How you would represent the image as a graph? What are the vertices and how are they connected?
   You can draw a picture if you’d like.

2. What kind of graph it is? For example, directed/undirected, weighted/unweighted, cyclic/acyclic, etc.

3. What graph processing algorithm would you apply to find the seam?

4. Would you need any special vertices that weren’t in the original image?

5. If you wanted to reduce the height of the image, instead of narrowing its width, how would your graph change?

12 Photoshop Flood Fill
15 points

Photoshop has a magic wand tool that allows you to click on a pixel in an image in order to select all of the pixels that share the same color that are adjacent to one another. This lets you turn a gray sky blue with a flood fill. Describe a graph processing algorithm that would let you implement the ability to select all adjacent pixels that are colored gray. In your explanation say how you would represent the image as a graph. What are the vertices? When do they have an edge? Is it a directed or undirected graph? Would it be preferable to use DFS or BFS for a very large image?

13 Finding the rank of an entry
10 points

Assume you have a BST and a min-heap and you are given an entry that exists in both data structures. You want to find the rank of the entry. An item is of rank \( k \) if it is the \( k \)th smallest value in its data set.

1. In what runtime could you find the rank of the entry in the BST? Why? (Note: you do not need to provide a full algorithm).

2. Assume that you changed the BST so that each node contained the total count of all of its descendants. How would that change the runtime?

3. Can you find the rank of this element in the same runtime in your min-heap? Why or why not?

14 Dijkstra’s Mod
10 points

Consider a graph with integer edge weights bounded by some positive integer \( k \). Describe a modification (to the priority queue) of Dijkstra’s algorithm to improve the complexity to \( O(V + E) \).