**Computer Science II (COP 3503)**

**Summer 2010 Final Exam**

**8/5/10**

**Lecturer: Arup Guha**

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1) (12 pts) Consider the following problem:

You have a set of n books with weights w1, w2, … , wn (in pounds). You must pack them into k boxes. The capacity of each of these boxes (in pounds) is c1, c2, …, ck. Our goal is to determine whether or not there exists a way to pack all of the n books in the k boxes without exceeding the capacity of any of the boxes. Assume that all of the book weights are positive integers less than 50 and all of the box capacities are less than 200. Furthermore, assume that both n and k are less than 100.

A proposed solution to this problem is to set up a Network Flow graph. Each book would have a vertex and each box would have one also. A source vertex, s, would be set up and a sink vertex t would also be added to the graph. An edge from s to each book would be set up with the capacity of that book. An edge from each book to each box would be set up with the capacity of the corresponding book. Finally, an edge from each box would be set up to the sink with the capacity equal to the maximum weight the box could handle. Determine the maximum flow of this network. If it is equal to the sum of the weights of the books, the books can be packed.. Otherwise they can not be.

What is the flaw in this solution? Create a simple example that shows that this solution does NOT work in all cases.

2) (12 pts) Determine the edit distance between the strings “COMPUTER” and “CHAPTER”. Complete the chart below. To get full credit you must fill it in properly. The table has been initialized for you with the edit distances between the empty string and all prefix strings.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | C | O | M | P | U | T | E | R |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| C | 1 |  |  |  |  |  |  |  |  |
| H | 2 |  |  |  |  |  |  |  |  |
| A | 3 |  |  |  |  |  |  |  |  |
| P | 4 |  |  |  |  |  |  |  |  |
| T | 5 |  |  |  |  |  |  |  |  |
| E | 6 |  |  |  |  |  |  |  |  |
| R | 7 |  |  |  |  |  |  |  |  |

3) (12 pts) In the jumble problem, you are given a set of letters, in random order, and you have to determine whether or not there exists a way to reorder those letters to form a valid word. One way to solve this problem is to generate all permutations of the input letters and check each permutation against a dictionary of words. Given that you have access to a function which tells you whether or not any words start with a set of letters, (for example startswith(“jx”) would return false while startswith(“po”) would return true), explain how you can solve the jumble problem using backtracking. Furthermore, explain why this solution is more efficient than the permutation solution.

4) (5 pts) The following list is a list of all the TV shows you want to watch and when they are on. Determine the most number of shows you can watch, assuming that you can only watch one show at a time and that when you watch a show, you watch it in its entirety. In doing so, utilize the greedy algorithm shown in class and fill out your TV watching schedule. (Note: You may not use all the slots given.)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Show | Time | Show | Time | Show | Time |
| Idol | 6:00-9:15pm | Survivor | 8:45-9:45pm | 24 | 7:30-10:30pm |
| CSI | 7:30-8:30pm | Office | 8:00-9:00pm | Law&Order | 9:50-10:50pm |
| Cops | 6:30-7:30pm | 30 Rock | 7:00-9:00pm | SportsCenter | 9:15-11:00pm |

|  |  |
| --- | --- |
| Show | Time |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

5) (12 pts) Create a minimum heap out of the following items by first placing them in a tree in the specified order and then executing the “makeHeap” function: 19, 17, 3, 22, 87, 14, 35, 4, 1, 26, 13, 67, 12, and 6.

Put a box around your final answer and show intermediate steps.

6) (15 pts) Determine the sum .

7) (12 pts) Examine the code below and answer the questions that follow it.

class dim {

public int r;

public int c;

public dim(int x, int y) {r = x;c = y;}

}

public class matchain {

public static void main(String[] args) {

dim[] a = new dim[5];

a[0] = new dim(2,4);a[1] = new dim(4,2);a[2] = new dim(2,3);

a[3] = new dim(3,1);a[4] = new dim(1,4);

System.out.println(function(a,0,4));

}

public static int function(dim[] all, int start, int end) {

if (start == end)

return 0;

else if (start + 1 == end)

return all[start].r\*all[start].c\*all[end].c;

int best = -1;

for (int m=start; m<end; m++) {

int one = function(all, start, m);

int two = function(all, m+1, end);

int three = all[start].r\*all[m].c\*all[end].c;

if (best == -1)

best = one+two+three;

if (one+two+three < best)

best = one+two+three;

}

return best;

}

}

a) What problem does this code solve? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

b) Symbolically, what does the variable m, represent?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

c) Explain conceptually what is stored in the variables, one, two and three.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

d) Is the method “function” efficient? Justify your answer.

8) (5 pts) Imagine running a bucket sort on a list of 20,000 items in the range [500,1000). In which bucket would the value 600.08 be placed? (Note: The buckets are numbered 0 to 19,999.)

9) (10 pts) At the end of running Floyd-Warshall’s algorithm on a graph with 10 vertices (labeled 0 through 9), one row of the path array appears as follows:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Value | 2 | 3 | 6 | 3 | 8 | 1 | 7 | 3 | 5 | 7 |

a) Which row (0 – 9) is this in the path array? \_\_\_\_\_\_

b) Determine the vertices traversed (and the order in which they are traversed) in the shortest path from vertex 3 to vertex 0, using the information given.

3 → \_\_\_ → \_\_\_ → \_\_\_ → \_\_\_ → \_\_\_ → \_\_\_ → 0 (Note: some of these may be left blank)

c) From the information given can we determine which vertex is farthest away from vertex 3? Why or why not?

10) (5 pts) What diary product flavors the popular Cheez-It crackers? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Scratch Page – Please clearly mark any work on this page you would like graded.**