Jacob Bollinger CS 112\* Final Exam

Answer each problem. Each of the 15 parts counts 4 points.

A design requires a paragraph format.

0) Given the real parameters a, b and c where a ≠ 0 for the quadratic equation

a · x² + b · x + c = 0, design a function to return a tuple of its roots.

Use a paragraph format.

Initialize roots to be a tuple and index to 0. Create a for loop to run for the length of the list. In the for loop append index to roots. Increase index by one.

1) Describe precisely and succinctly what the following function does

where aList is a list.

def f( aList ) :

looking = True

index = 0

while looking and ( index + 1 < len( aList ) ) :

looking = aList[ index ] not in aList[ index + 1 : ]

if looking :

index = index + 1

return looking

This function checks if there are any repeated elements side by side and returns True if there are none.

2) Design a function to find the grade point average of a student's courses

for one semester given as a space separated string of

departmentAbbreviation courseNumber credits grade.

For example, the string

'ENG 111 3 B MATH 131 3 A CS 112 3 B CS 112L 1 A PHIL 210 3 B'

yields a grade point average of 3.54.

where A has 4 quality points, B 3, C 2, D 1, F 0. A grade of W does not

count in the computation of grade point average.

Use a paragraph format.

Start by naming the string studentCourses. Then strip and split the string. Set a variable named index equal to 3 and gpa equal to 0. Create a for loop that runs for the length of the list divided by four. In the for loop create an if statement to test if studentCourses[index] is equivalent to ‘A’. if it is, add four multiplied by int(studentCourses[index – 1]) to gpa. Create four else if statements to test if the value of studentCourses[index] is equal to ‘B’, ‘C’, ‘D’, or ‘F’. If it is ‘B’, add three multiplied by int(studentCourses[index – 1]) to gpa. If it is ‘C’, add two multiplied by int(studentCourses[index – 1]) to gpa. If it is ‘D’, add one multiplied by int(studentCourses[index – 1]) to gpa. Outside of the if statements, increase index by four. Outside of the for loop set gpa equal to gpa divided by the length of studentCourses divided by four.

3a) Describe the binary search algorithm for a value of key in a sorted list.

b) Simulate the binary search algorithm for key of 'bart' in list of

[ 'a', 'ad', 'am', 'an', 'at', 'ax', 'bat', 'be', 'bed', 'blue', 'buy', 'by' ]

0 1 2 3 4 5 6 7 8 9 10 11

Use a table with appropriately labeled columns.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | list[value] | key < | key > | key = |
| bart | ax | False | True | False |
|  | bed | True | False | False |
|  | bat | True | False | False |
|  |  |  |  |  |

4a) Describe the merge algorithm for 2 sorted lists.

For oldList1 and oldList2 merge will combine these in order of elements. This means that it will compare oldList1[ndx1] and oldList2[ndx2] to see which value is less and then add it to the list. It will then increase the corresponding ndx by one and compare the lists once again. This continues until it has cycled through all values by oldList1 and oldList2.

b) Design an efficient algorithm to add 2 polynomials, where a

polynomial is represented as an ordered list of its coefficients.

The polynomial p = 3x³ - 2x + 4 has the representation [ 4, -2, 0, 3 ].

The polynomial q = -2x⁴ + 6x³ + 5x² + 7x has the representation

[ 0, 7, 5, 6, -2 ].

The polynomial p + q = -2x⁴ + 9x³ + 5x² + 5x + 4 has the representation

[ 4, 5, 5, 9, -2 ]

Use a paragraph format.

Start by initializing the list mergedList. Then determining whether list p or q is longer and make p the longer list. Create a for loop that runs for i in range of p. In the for loop append p[i] plus q[i] to mergedList.

c) Provide pseudo code for your algorithm of part b.

mergedList = []

if (len(q) > len(p)):

p, q = q, p

for i in range p:

mergedList.append(p[i] + q[i])

5a) Describe Python's structure called dictionary.

A dictionary is a list of keys and correlating values.

b) The registrar has a text file where each line consists of

studentID, courseID, grade.

Design an efficient algorithm to produce the grade distribution for

each course. A grade distribution for a course consists of a count

of its number of As, Bs, Cs, Ds, Fs and Ws.

Use a paragraph format.

c) Provide pseudo code for your algorithm.

6a) Define an algorithm.

An algorithm is a length of code that completes a process.

b) Describe the role of algorithms in CS 112. Illustrate your remarks with

examples from the course and the lab.

We have used many algorithms throughout this course and in each lab. In lab 0, we used algorithms to convert metric and imperial units. In lab eight we used algorithms to sift through data. In lab 9 we used algorithms to give robots strategies.

7a) Design a efficient algorithm with a sorted list parameter that returns

a list of the item or items that appear most often.

For example, the list [ 3, 8, 8, 9, 9, 9, 10, 11, 12, 14, 14 ] returns

the list [ 9 ] as its most frequent item.

For another example, the list [ 0, 0, 1, 2, 2, 3, 4, 5 7 ] returns the

list [ 0, 2 ] as its most frequent items.

Use a paragraph format.

b) Provide pseudo code for your algorithm.

frequency = 0

mostFrequent = []

for i in sortedList:

if(sortedList[i] == sortedList[i-1]):

increase frequency by one

elif frequency > oldFrequency:

mostFrequent.append((sortedList – 1)

oldFrequency = frequency