STAT 598Z Midterm

March 08, 2012

Time: 75 minutes

Your Name (Please print): _____

PUID (Please print): _____

Note:

1. This exam will contribute 10 points towards your final score
2. Use the provided scratch paper for your calculations
3. Take a print out of your code and attach it along with all relevant calculations
4. **Write your name clearly and legibly on all printouts**
5. Attempt as many problems as possible and explicitly state all your assumptions.
6. Show all intermediate steps for full credit. Python code must be clear and concise.
7. Any attempt at academic dishonesty (e.g. using a browser during the exam) will automatically result in 0 points.
8. Use of notes, books, laptops, cell phones, or any other aids (electronic or otherwise) is strictly prohibited. Turn off and put away your cell phone now!

Please sign below to indicate your agreement with the following honour code.

Honour code: I promise not to cheat on this exam. I will neither give nor receive any unauthorized assistance. I promise not to share information about this exam with anyone who may be taking it at a different time. I have not been told anything about the exam by someone who has already taken it.

Signature:_____ Date:_____

Questions	Possible Points	Actual Points
1	2	
2	3	
3	2	
4	3	
Total	**10**	

Problem 1 (2 pt) Recall that a palindrome is a string which spells the same if its spelt from front or back. For example `madam` is a palindrome. Write two Python functions `palindrome_for` and `palindrome_which` which use a for loop and a while loop respectively to check if a given input string is a palindrome. Your functions must take a string as input and return True if the string in a palindrome and False otherwise. Print the output of both your functions on the following strings:

- `madam`
- `amma`
- `abcde`
- `aradar`

Solution 1:

Problem 2 (1.5 + 0.5 + 1 pt) Recall the mergesort algorithm we discussed in the class. One of the key components of mergesort was a sub-routine for merging **two** sorted lists.

- Write a python function `threeway_merge` which takes as input **three** sorted lists `a`, `b`, and `c` and merges them to produce another sorted list `d`.
- What is the time complexity of your algorithm in O notation. Provide clear and concise arguments.
- Your friend claims that her mergesort algorithm which is based on `threeway_merge` is asymptotically better than the standard mergesort based on merging two lists at a time. Do you agree? Justify your stand in a mathematically precise way by using the O notation.

Solution 2:

Problem 3 (1 + 1 pt) Devise a recursive algorithm for finding the sum of the first n **odd** positive integers.

- write down the pseudo-code of your algorithm.
- State the recurrence, draw the recursion tree, and find the complexity of your algorithm with respect to n in Θ notation.

Solution 3:

Problem 4 (3 pt) Generate samples from the triangular pdf

$$p(x) = 2x \text{ for } 0 \leq x \leq 1 \tag{1}$$

using **two** different methods. Write Python functions to verify your algorithms by generating 10,000 samples from the above distribution and plotting their histogram. Describe your scheme in the space below, and attach printouts of both source code and the plotted histogram for full credit.

Solution 4: