CSC 106 Assignment 2: Sorting through the meaning of Life(Life), and in through the Or Gate

Due: Fri, Feb 10, 2012 at 4:30 pm

Marks: **50 marks.**

Learning Goals:

At the end of this assignment you will be able to:

- Describe the QuickSort method of sorting a list of items;
- Determine which sort might work best in a given application;
- Design and create a circuit diagram using logic gate symbols;
- Design a starting pattern for the game of life ;

Submission: This assignment is to be done individually. The submitted assignment can be either a .doc, .docx, .pdf, .rft, or .pages file. It should be attached to the CSC106 Assignment 2 submission page on Connex.

A few reminders:
---- Make sure that you include your NAME, STUDENT #, and email in the Assignment Text box on Connex. As well, please make sure that your NAME and STUDENT# are on the submitted file itself.

-- Please make sure that you include the Task number/section you are answering above your response. This makes it easier for your marker to stay with the flow of your answers.

-- Make sure that your file is attached before you click the final submission button on Assignment 1 in Connex. There is a bug in the Connex submission attachment, so you only get one chance to submit you work. Make sure it is all there before you press the final submit.

Resources:

- The Game of Life - Scientific American 223 (October 1970): 120-123.

http://ddi.cs.uni-potsdam.de/HyFISCH/Produzieren/lis_projekt/proj_gamelife/ConwayScientificAmerican.htm (if this link doesn't click properly, copy and paste it into a browser).

- The Game of Life - online http://www.bitstorm.org/gameoflife/
- Conway's Game of Life on Wikipedia:

http://en.wikipedia.org/wiki/Conway%27s_Game_of_Life

- Video overview of various sorts (Quicksort is towards the end -- spoiler alert - they don't like it very well!): http://youtu.be/INHF_5RIxTE
- Overview of a number of sorts with animations (the descriptions are a bit

technical, but the animations are good):
http://www.sorting-algorithms.com/

TASK 1: Sorting (15 marks)

How does the sorting algorithm Bubble Sort work?

1. Research this algorithm and describe it *in your own words* (not copied from the research materials you found). Do not give pseudo code, but rather an overview of why and how the algorithm works. Feel free to use drawings or examples to support your answer. Provide references to any material you used to help come up with your description
2. Give pseudo code for Bubble Sort that matches your description in (1).
3. Consider this list: 10, 1, 3, 9, 8

If you used Bubble Sort to sort this list into ascending order how many comparisons in total would it take? How many swaps would it take?

What if the list was 10, 9, 8, 3, 1

How many comparisons in total would it take to sort this list using Bubble Sort? How many swaps would it take?

TASK 1 DELIVERABLES: A word processed document containing the Task heading, the description (written in your own words), any diagrams you wish, and references/citations listed below the description. The pseudo code should be below the description.
Marks will be assigned as follows:

5 Marks for 1. (Remember to give your references in one of the styles discussed in lab. Be as clear as possible in your description. Don't copy and paste -- must be in your own words.)

5 Marks for 2. (Remember, state assumptions.)

5 Marks for 3. (I have not asked you to show your work, but don't just get the answer from someone else. This would be a great type of question for a midterm or final -- hint, hint -- so you should know how to approach a question like this. If you get help from someone, make sure you see how the comparisons and swaps work.)

(This Task is worth 15 marks).

TASK 2: Picking the right sort of sort. (5 marks)

You are part of a software design team creating a game for a mobile device (e.g. iPhone, tablet, iPad, etc.). It is a fantasy role playing game. The goal of the game is to develop

characters of sufficient power so that the characters can that can take on ever more difficult quests. Any quest that a character undertakes will involve expending and replacing (hopefully) that character's power.

The game arena contains objects, and each object has an amount of power associated with it. When characters obtain these objects, the power associated with the object transfers to the character. Power allows characters to move about the fantasy realm, create structures and tools, overcome other characters, etc. -- essentially anything that helps the character solve the current quest.

The game space is full of power objects. Each object has an arbitrary power "value", so some objects are more powerful than others. Each character begins with 1 object. Each character can amass up to 10 objects. Every time a character gathers a new object a sorted list of that character's power objects is presented on the game screen, with the new object correctly inserted. It is a linear list, sorted in order of least powerful to most powerful. Any time a character expends power and uses up the power of an object (e.g. in a battle, forging a new sword, etc. power would be expended and objects would be used up and would not be considered the next time the sort was conducted), or picks up a new object the list is sorted again and presented on the game screen. As well, a sorted list can be requested by its owner any time throughout the game.

Your task is to select an algorithm to handle the sorting. At this point you don't have to worry about coding, or defining data structures, but you do have to select a sorting algorithm to use at the back end. You narrow your selection down to two different comparison sorting algorithms that you think will do the job. You select two comparison sorts - Insertion Sort and Quick Sort

Create a table comparing the two algorithms. Make sure to compare how objects with the same power value are handled. As well, compare space requirements, worst case number of comparisons, and number of swaps. As a final point, compare the ease of programming the two different sorts.

Finally, write no more than a paragraph noting which one of the two you recommend and why.

TASK 2 DELIVERABLES:
Using the same word processed document as you created for Task 1 make a new Task 2 heading.

3 marks for correctly comparing two algorithms.

2 marks for well reasoned final recommendations.

(This Task is worth 5 marks).

TASK 3: Logic Gate Design (20 marks)

Implement this sequence of gates:

((A AND B) OR (C AND NOT(A))) AND (A AND D)

1. Draw a diagram of this circuit using the logic gate symbols shown in the lab.

2). Fill in a truth table for this circuit

Use the notes from the 2nd lab to help you create the circuit. If you like, you can draw (neatly!) the circuit on paper, take a photo of it, or scan it, and insert it.

TASK 3 DELIVERABLES: Responses to this question in your word processed document. PDF format is preferred, but just about any word processed file format will work. Use the notes from the 2nd lab to help you create the circuit. If you like, you can draw (neatly!) the circuit on paper, take a photo of it, or scan it, and insert it into your word processed document.

5 marks for correct response in the table.

15 marks for correct diagram.

This Task is worth 20 marks

TASK 4: Game of Life startup (5 marks)

Using any of the Game of Life applications
(e.g. http://www.bitstorm.org/gameoflife/ to play online
http://golly.sourceforge.net/ if you want to download a game)

1. Using between 5 and 10 "live" cells as your start point, create a layout that either stabilizes (becomes still) or oscillates. Paste a copy of your initial layout below. You just need to paste in the block of cells that immediately surround and include your initial layout.
2. Who was John Conway and why did he develop the game of life? (No more than one paragraph please. Cite your reference.)

TASK 4 DELIVERABLES: Using the same word processed document as you created for Task 1 make a new Task 4 heading and paste a copy of your starting Life grid below it. Answer the John Conway question below the Life grid.

3 marks for a still life or oscillating pattern.

2 marks for the Conway question.

This Task is worth 5 marks.

OVERALL ASSESSMENT (5 Marks):
Five marks are set aside for overall assessment. 4 to 5 marks are given for outstanding work. The assessment proceeds as follows:

5 marks given for an outstanding assignment - original thought, outstanding organization, well demonstrated analysis and synthesis capabilities, excellent grasp of subject matter with sound critical evaluations throughout.

4 marks given for strong evidence of original thinking; excellent organization; capacity to analyze and synthesize; superior grasp of subject matter with sound critical evaluations. All other material in the assignment is correctly completed in an exemplary manner.

1-3 marks given for evidence of grasp of subject matter, some evidence of critical capacity and analytic ability; reasonable understanding of relevant issues; evidence of familiarity with the concepts. All other material in the assignment is correctly completed.
Final note -- the staff at the **Computer Science Consultants Office (Help Desk)** are there to help you. They can't "do" the assignment for you, but they can help out with application problems, printer problems, understanding questions, saving documents, uploading and moving files, etc. Call on them for help.