**CSE 3302: Programming Languages**

**Spring 2018**

**Homework 05**

**Due on October 12, 2018 [ before 11:59 pm]**

**Name: Jerry Olds**

**1000 number: 1001533643**

**Date: 10/07/2018**

**INSTRUCTIONS**

1. **Do NOT plagiarize.**
2. **No group-work. All work should be your own.**
3. **Do not discuss your work with other students in the class.**
4. **Cite sources where necessary.**
5. **Turn in your word document using Blackboard. Do not email your documents.**
6. **Name your document as netid.docx where *netid* is your UTA NetID. If you do not know your NetID, check what it is using NetID Self Service. Your 1000 number is NOT your NetID.**
7. **Try to answer each question within 5-7 lines.**

**Questions:**

**1.**What consists a good algorithm? **[7 points] Fewer amount of steps to compute and less memory usage.**

**2.** Why the documentation is important for software engineering?**[8 points] Documentation is important because software is constantly changing and being maintained. If all the changes to software are documented, it helps the developers create an overall better product.**

**3.** List 7 different data structures. **[7 points]**

**Array, Linked List, Stack, Queue, Binary Tree, Heap, Hash Table.**

**4.** Explain what are the space complexity and time complexity in one algorithm.**[8 points] Time complexity is the amount of time an algorithm will take to complete its task. Space complexity is the amount of memory it will use to complete the task.**

**5.** Why do we need a right data structure when solving problems.**[5 points] Different data structures are useful when it comes to certain computations. Using the right data structure in the right situation makes the developer’s job a lot easier.**

**6.** Compare queue with stack. How will you implement them use array and linked list?**[10 points] Queues are First-In First-Out (FIFO), Stacks are Last-In First-Out (LIFO). To implement a queue with linked list in array, you create a pointer to the array or linked list. To insert a node in to a queue you traverse the linked list or array and insert it at the end. To implement a stack you create a tree with a linked list or array and each node that is inserted becomes a parent.**

**7.** What is Object Oriented Programming?**[5 points] Idea of packing up functional units into nested objects.**

**8.** What is a circular linked list?**[8 points] When the last node of a linked list points back to the first node.**

**9.** What is a Turing machine?**[5 points] Theoretical computing machine that can simulate any algorithm given an infinitely long memory tape, a state variable, and rules to instruct the machine what to do given the state and what is read from the tape.**

**10.** Do you think the private function in C++ is important? Why? **[7 points] Yes, I believe the private function is important. If a member variable or function is not going to be used anywhere else besides in the class that it was created, there is no reason to make it public. If an error occurs with a certain variable or function it is much easier to identify the issue if it is private because you know the error is somewhere within the class that created it. If it is public the issue can be anywhere that has access to the class, making it much harder to identify. Making variables and functions public when they don’t need to be creates the opportunity for more errors to occur.**

**11.** What is the big O annotation? **[7 points] A way to write the complexity of an algorithm. It shows relationship between input size and number of steps the algorithm takes to solve.**

**12.** Sort the time complexity of O(n), O(nlogn), O(n^2), O(n + 5\*log n), O(3n + 5\*log(logn))**[7 points]**

**O(nlogn), O(n), O(n + 5\*log n), O(3n + log(logn)), O(n^2) (least to greatest).**

**13.** Compare the time complexity of quick sort, merge sort and bubble sort in worst, best, average cases. Which one do you prefer? Why?[**10 points]**

**Best Average Worst**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quick** | **O(nlog(n))** | **O(nlog(n))** | **O(n^2)** |
| **Merge** | **O(nlog(n))** | **O(nlog(n))** | **O(nlog(n))** |
| **Bubble** | **O(n)** | **O(n^2)** | **O(n^2)** |

**In general, I prefer the merge sort out of these three because it’s easy to comprehend and the most efficient. The bubble sort is the easiest to comprehend and the easiest to implement so if efficiency is not an issue, I would use a bubble sort. In most cases where efficiency is an issue I would use the merge sort.**

**14.** What is a brute force approach?**[6 points] It’s an exhaustive approach in which all the possible outcomes are used until the desired result is found.**

**Extra credit (bonus question):**

**15.** List 5 different sorting algorithms and give a brief introduction to them. **[10 points]**

**Bubble Sort: Compares two elements of array and swaps them if they are in the wrong order. It does these comparisons until it reaches the end of the array and then starts from the beginning until there are no swaps made.**

**Merge Sort: Splits an array in to two halves, sorts the two halves, then merges the two halves in to one sorted array.**

**Quick Sort: Picks an element of an array as a pivot and partitions the array around the pivot.**

**Insertion Sort: Takes one element of an array at a time and inserts each element in to its proper position.**

**Heap Sort: Takes advantage of the Heap structure in which it sorts subtree by subtree by finding the maximum value and putting it at the end of the subtree.**