|  |  |
| --- | --- |
| **CSCI 2100E Spring 2020** | Student Name: \_Tam Yi Ki\_\_ Student Id: \_\_1155126127\_ |

# "Written" Exercise #10

|  |
| --- |
| Please give answers to this exercise via writing or drawing when needed.  This exercise is counted as a regular lab exercise in terms of weighting.  Aim of this exercise is to give you more experience in answering conceptual questions in data structures topics.  In these concept questions, your answers are merited by the proper use of terminology and demonstration of understanding of subject matter. In both exams and exercises, answer these questions as if you are attending an industry test/interview, as it is assessed as such. |

## Question 1 Binary Expression Tree

We have a arithmetic expression (4 + 5) \* (9 - 7) / 2 + 6 - 1

a) Please construct a Binary Expression Tree (BET) based on the above expression. You can draw on paper and photo/scan it or use vector graphics features in your word processor.

|  |
| --- |
|  |

b) Using the answer in a), derive and write down the polish and reverse polish notation of the expression.

|  |
| --- |
| Polish: + / \* + 4 5 – 9 7 2 – 6 1  Reverse polish: 4 5 + 9 7 - \* 2 / 6 1 - + |

c) We now want to model the unary operator *negation*, e.g. -1. Using your knowledge in Binary Expression Tree, deduce what the BET would be like for the expression (-(-(-3))) \* (-(-2))

|  |
| --- |
| 一張含有 相片, 就緒, 白色, 黑色 的圖片  自動產生的描述 |

## Question 2 Binary Search Tree

We now have a series of seven integers: 30, 66, 23, 1, 100, 15, 91

a) Assume we insert the above integers into a Binary Search Tree (BST) in the order shown above, what would the BST be like after each insertion? Please indicate the step clearly.

|  |
| --- |
| 一張含有 物件, 時鐘, 手錶 的圖片  自動產生的描述  30 is the root  66 is greater than 30, so it is the right child of 30  23 is smaller than 30, so it is the left child of 30  1 is smaller than 23, so it is the left child of 23  100 is greater than 66, so it is the right child of 66  15 is greater than 1 and smaller than 23, so it is the right child of 1  91 is smaller than 100 and greater than 66, so it is the left child of 100 |

b) Perform a pre-order and post-order traversal of the BST in a) and write down the print orders

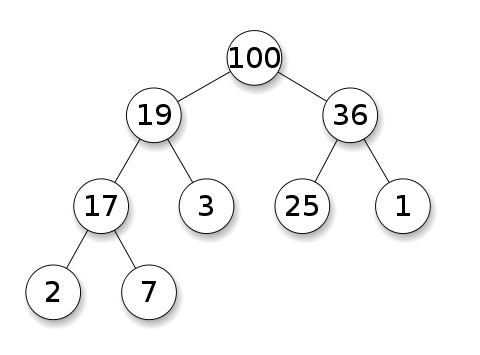
|  |
| --- |
| Pre-order: 30 23 1 15 66 100 91  Post-order: 15 1 23 91 100 66 30 |

c) Draw a completely unbalanced but valid BST with the same seven integers. You do not need to show any steps.

|  |
| --- |
| 一張含有 物件, 時鐘 的圖片  自動產生的描述 |

## Question 3 Heap

We have a binary heap as shown. It is a max-heap, i.e. the children of a node must be smaller than its parent.



https://commons.wikimedia.org/wiki/File:Max-Heap.svg

a) Please show the array representation of the heap, if we adopt the implementation in the lecture notes.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Content | 100 | 19 | 36 | 17 | 3 | 25 | 1 | 2 | 7 | | Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | |

b) If I insert/enqueue key 18 into the heap in a), what would the heap be like? Please provide both the tree representation and array implementation.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 一張含有 掛, 光, 白色, 電線 的圖片  自動產生的描述   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Content | 100 | 19 | 36 | 17 | 18 | 25 | 1 | 2 | 7 | 3 | | Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |

c) If I withdraw the root 100 from the heap in a), what would the heap be like? Please provide both the tree representation and array implementation.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 一張含有 尋找, 白色 的圖片  自動產生的描述   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Content | 36 | 19 | 25 | 17 | 3 | 7 | 1 | 2 | | Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |

d) Draw a valid min-heap with the exactly the same set of integer keys in a)

|  |
| --- |
| 一張含有 室內, 掛, 電線, 尋找 的圖片  自動產生的描述 |

## Question 4 Set ADT by Binary Search Tree

I have a set of integers 201, 901, 550, 300, 700, 101, 444. I want to create a Set ADT that keeps track of this set of integers. In reality, many engineers will choose to choose a Binary Search Tree to implement the Set ADT.

a) Assume now I am inserting the keys into the Set ADT in the listed order above, what would the Binary Search Tree be like?

|  |
| --- |
| 一張含有 室內, 螢幕, 掛, 白色 的圖片  自動產生的描述 |

b) What would be the main rationale of the common practice of using a Binary Search Tree over a simple binary tree to implement the Set ADT?

|  |
| --- |
| Faster |

c) Name one main advantage and one main disadvantage of using a Binary Search Tree over hashtable to implement a Set ADT?

|  |
| --- |
| Advantage: You have to know approximate size of input data before initializing hash table. Otherwise you need to resize hash table which is a very time-consuming operation.  Disadvantage: Binary search tree never meets collision, which means binary search tree can guarantee insertion, retrieve and deletion are implemented in O(log(n)) |