**Mini review 1**

1. Create an abstract of Complex number data type.
2. Given a singly linked list Head containing *n* integer numbers, write two functions to do the following tasks:

* Find the smallest odd number of Head
* Remove all odd numbers of Head

1. Sort the following functions in the increasing order of Big O notations:

|  |  |  |
| --- | --- | --- |
| 4nlogn + 2n | 210 | 2logn |
| 3n+100logn | 4n | 2n |
| n2 + 10n | n3 | nlogn |

1. Given an integer number *n*, your task is to write two different algorithms in pseudo-codes to calculate 2n, and evaluate the complexity of the algorithms.
2. Calculate the complexity of the following functions:

**Function sum:**

sum = 0;

for ( i = 0; i < n; i + +)

for ( j = i + 1; j < = n; j + +)

for ( k = 1; k < 10; k + +)

sum = sum + i \* j \* k ;

**Function Matrix:**

for (i = 0 ; i < n ; i++)

for (j = 0 ; j < n ; j++)

if (i == j)

A[i][j] = 1;

else

A[i][j] = 0;

1. Given a list of students (id, name):

(7,An), (3,Be), (11, Cu), (4, Da) , (8, Gi), (16, En), (21, Ba), (5, Go)

Your task is to propose a hash function, and draw the hash table with the proposed hash function using both collision handling methods

1. Do following tasks with the heap tree:

* Construct a max heap tree including: 2, 19, 38, 29, 66, 64, 72, 3, 16, 89, 15, 37, 20, 28, 73, 5.
* Insert the following numbers into the above max heap tree: 5, 13, 9, 7, 24, 4, 6

1. Do the following tasks with binary search tree:

* Create a binary search tree from following numbers: 14, 15, 35, 62, 29, 42, 40, 80, 59, 23, 46, 57, 3, 19
* Draw BSTs after deleting keys 12, 42 and 13 from the above tree.

1. write *TreeSearch*(*k*, *T*) function to check if *k* is in the binary tree search *T*.
2. Find the order of nodes in preorder, postorder, and inorder traversals





