CSC 382 Analysis of Algorithms

Exam I

1. (3 points) **Power of Two**

   Given an integer number, write a recursion function return `true` if the number is a power of two. Otherwise, return `false`.

2. (3 points) **Insert Function of a Vector**

   Vectors (or **Dynamic Arrays**) are sequence containers representing arrays that can change in size.

   Write the `insert` (insert a new element at a specified position) function of a vector.

```
-- Example --
  cout << vec << endl;   // output: 3, 1, 4, 2, 8
  vec.insert(5, 2);      // insert 5 at index 2
  cout << vec << endl;   // output: 3, 1, 5, 4, 2, 8
```

3. (3 points) **Two Sum**

   Given an array of integers `nums` and an integer `target`, return indices of the two numbers such that they add up to `target`.

   You may assume that each input would have exactly one solution, and you may not use the same element twice.

   You can return the answer in any order.

```
-- Example --
       Input: nums = [2,7,11,15], target = 9
      Output: [0,1]
 Explanation: Because nums[0] + nums[1] == 9, we return [0,1].
```

4. (3 points) **Convert Sorted Array to Binary Search Tree**

   Given an integer array where the elements are sorted in ascending order, convert it to a balanced binary search tree.

5. (3 points) **Last k-th of List**

   Find the kth node from the end of a singly linked list.

```
-- Example --
        Input: head->(3)->(1)->(4)->(2)->(8)->(9)->(7)->(6), k = 3
       Output: (9)                              ^
```

6. (3 points) **Find Peak Element**

   A peak element is an element that is strictly greater than its neighbors.

   Given an integer array, find the peak element.

   Assume the peak appears somewhere in the middle and there is only 1 peak in the array.

```
-- Example --
        Input: [1,2,3,5,8,9,7,6,4,2]
       Output: 5          ^
```

7. (3 points) **Last Stone Weight**

   You are given an array of integers `stones` where `stones[i]` is the weight of the `i`-th stone.

   We are playing a game with the stones. On each turn, we choose the heaviest two stones and smash them together. Suppose the heaviest two stones have weights `x` and `y` with `x <= y`. The result of this smash is:

   - If `x == y`, both stones are destroyed, and

   - If `x != y`, the stone of weight `x` is destroyed, and the stone of weight `y` has new weight `y - x`.

   At the end of the game, there is at most one stone left.

   Return the weight of the last remaining stone. If there are no stones left, return `0`.

```
-- Example --
        Input: stones = [2,7,4,1,8,1]
       Output: 1
  Explanation:
      We combine 7 and 8 to get 1 so the array converts to [2,4,1,1,1] then,
      we combine 2 and 4 to get 2 so the array converts to [2,1,1,1] then,
      we combine 2 and 1 to get 1 so the array converts to [1,1,1] then,
      we combine 1 and 1 to get 0 so the array converts to [1].
```