CSE 214
Homework #3
Binary Search Tree, Heaps, 2-3-4 tree

Homework guideline:
Solve the problems using binary search tree/heaps/2-3-4 tree. Give optimal solution.

Submission guideline:
Submit the homework through blackboard. Before submission make sure your codes do not have any error, unexecutable code and/or late submission will not receive any credit. Submit your Java code (.java files only) as a single .zip archive. Include a README text file to give the TA's instructions on how to run your code. The .zip file name should be in the following format: < firstname >_< lastname >_< id > _hw< num >.zip
For example, if John Doe with student ID 123456789 is submitting the third homework, the submitted file should be named john_doe_123456789_hw3.zip

Number Distribution:

(1) 15
(2) 15
(3) 20

Total Marks: 50
Homework posted on: October 22, 2017
Submission Date: October 31, 2017 (11:59 PM)

(1) Given a complete binary tree with N nodes and each node have a distinct integer a_i attached with it, find the minimum number of swaps you can make to convert the binary tree into binary search tree. In one swap, you can select any two nodes and swap their values.

You will be given the array representation of the binary tree. Root of the tree will be at a_0. Left child of root will be at a_1 and right child of root will be at a_2. Left child of node at array position k will be at a_{2*k+1} and right child of node at array position k will be at a_{2*k+2}.

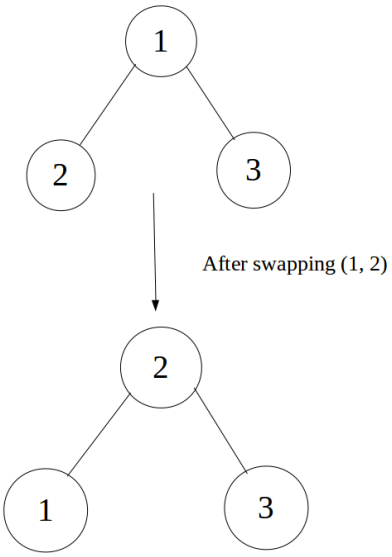
Input format:
First line contains an integer, N ($1 \le N \le 105$), denoting the number of nodes. Second line contains N space separated integers, a_i ($1 \le a_i \le 105$), denoting the value attached to ith node.

Output format:
Print a single integer, denoting the minimum number of swaps needed to convert binary tree into a binary search tree.

Sample Input:
3
1 2 3

Sample Output:
1

Explanation
We need only one swap (1,2) to convert the given binary tree into binary search tree.



(2) Sean is a big fan of Coldplay, which is a British rock band. Coldplay is in New York City for their "A Head Full of Dreams Tour". They will perform in the MetLife Stadium. So, Sean decided to go and watch his favorite band perform. After reaching the stadium, he saw that many people have lined up for the concert tickets. He knows that there are **M** rows in the stadium with different seating capacities. They may or may not be equal. The price of the ticket depends on the row. If the row has **K** (always greater than 0) vacant seats, then the price of the ticket will be **K** dollars. Now, every fan standing in the line will get a ticket one by one. Given the seating capacities of different rows, find the maximum possible amount that the concert event will gain with the help of the ticket sales.

Input:
The first line consists of **M** and **N**. **M** denotes the number of seating rows in the stadium and **N** denotes the number of Coldplay fans waiting in the line to get a ticket for the match. Next line consists of **M** space separated integers **X[1], X[2], X[3]...., X[M]** where **X[i]** denotes the number of empty seats initially in the i^{th} row.

Output:
Print in a single line the maximum dollars the concert event will gain.

Constraints:
1 <= **M** <= 1000
1 <= **N** <= 1000
1 <= **X[i]** <= 1000
Sum of **X[i]** for all **1** <= **i** <= **M** will always be greater than **N**.

Sample Input:
3 4
1 2 4

Sample Output:
11

Explanation

In the sample test case, number of rows is 3 and 4 people waiting in line to get a ticket. Since the maximum cost of ticket initially is 4 dollars, therefore the first person in line will buy a ticket for the 3rd row. The next person standing in line will again choose the 3rd row as it has the maximum number of seats, which will cost him 3 dollars. The next person will have 2 choices, he can either choose the 2nd row or the 3rd row which will cost him 2 dollars. Similarly, the last person will choose the row will 2 seats remaining, which will cost him 2 dollars. Total cost = 4+3+2+2 = 11 dollars.

(3) Design your own 2-3-4 tree class [see 2-3-4 tree lecture for details about the operations]. Your tree class should have the following operations,

(i) Insert
(ii) Delete
(iii) Search
(iv) Traverse
    (a) Inorder
    (b) Preorder
    (c) Postorder

Input Format:

The first line of the input contains Q, the number of operations. The next Q lines contain the operations that need to be performed. Following are the different types of operation,

(i) Insert: 1 k [where, k is the key to be inserted]
(ii) Delete: 2 k [where, k is the key to be deleted]
(iii) Search: 3 k [where, k is the key to be searched]
(iv) Traverse
    (a) Inorder: 4
    (b) Preorder: 5
    (c) Postorder: 6

Initially, the tree is empty.

Output format:

After each 4, 5, and 6 output the corresponding traversal. For successful search output "successful", otherwise output "failed". For successful delete output "successful", otherwise output "failed".

Sample Input:

```
12
1 3
1 2
1 6
1 1
4
1 8
1 9
1 7
1 10
1 11
4
3 12
```

Sample Output:

```
1 2 3 6
1 2 3 6 7 8 9 10 11
failed
```