UCS-406 (Data Structure)

Lab Assignment-1 (2 weeks)

Implement the following programs in C/C++/Python/Java using functions

a) Insertion Sort
b) Bubble Sort
c) Selection Sort
d) Linear Search
e) Binary Search
f) Shell sort

Q 1. Display the total number of comparisons and swappings made by each searching/sorting function for the given input N.

Q 2. Create a structure/class for a group of 50 students holding data for their Regn no., Name, Branch, CGPA

a) Call linear search function to display data of student with a particular Regn no..
b) Call bubble sort function to arrange data of students according to Regn no.
c) Apply binary search on the above output (part b) to display data of a student with a particular Regn no.
d) Use and modify Insertion sort logic to arrange data of students in descending order of CGPA.

LAB Assignment-2

Q 3. Write a program to perform following operations on stack.

a) Create functions for push and pop operations of stack.

b) Write a function to convert an infix expression to postfix expression. Pass a one dimensional character array P to the function as input (infix exp) and return character array Q (postfix exp). Test your program for following input

P : (A – (B / C) * D + E) * F % G

c) Write a function for the evaluation of a given postfix expression. For testing pass the postfix expression Q of part b and supply following set of values.

A = 90, B = 50, C = 2, D = 3, E = 1, F = 2, G = 5

Q 4. a) Write program using functions to implement Mergesort.

b) Write two different (recursive and non-recursive) functions to implement quick sort.

c) Write two different (recursive and non-recursive) functions to solve towers of Hanoi.

Lab Assignment-3

Q 5. Write a program using functions for implementation of circular Queue.

Q 6. Write a program to perform following operations on Link List

a) Insertion
   - at the beginning
   - at the end
   - at the given location
   - in the sorted list

b) Deletion
   - of first node
   - of last node
   - of given item of node
   - of given item from sorted list

c) Given a list, split it into two sublists — one for the front half, and one for the back half. If the number of elements is odd, the extra element should go in the front list. So FrontBackSplit() on the list {2, 3, 5, 7, 11} should yield the two lists {2, 3, 5} and {7, 11}.

d) Given two one-way-link lists A and B. Sort them independently and then Merge as list C.

e) Two way link list insertion and Deletion

f) Header link list insertion and deletion

Lab Assignment - 4

Q 7. Write program using functions for binary tree traversals Pre-order, In-order and Post-order using both Recursive and Non-recursive approach.

Q 8. Implement following functions for Binary Search Trees

a) Search a given item (Recursive & Non-Recursive)
b) Insertion of a new node
c) Maximum element of the BST
d) Minimum element of the BST
e) Successor of the BST
f) Delete a given node from the BST

Q 9. Implement Heap Sort.

Q 10. Implement Strassen Matrix Multiplication problem using Divide & Conquer approach.

Q 11. Implement following problems using Greedy Approach:
a) Partial Knapsack
b) Huffman compression

Lab Assignment - 5

Q12. Write a program to implement concept of hashing. Design a menu based interface to call different functions for collision handling techniques.

Implement the following problems for Graphs

Q13. Write a program to find approachable nodes from a given source of a given graph using queue as an intermediate data structure (BFS).

Q14. Write a program to traverse various nodes of a given graph using stack as an intermediate data structure (DFS).

Q15. Write a program to find shortest path from a given source to all the approachable nodes (Single source shortest path Dijkstra's algorithm).

Q16. Write a program to find shortest path between all the source destination pairs (All pairs shortest path Floyd's algorithm.

Q17. Write a program to arrange all the nodes of a given graph (Topological sort).

Q18. Write a program to find Minimal spanning tree of a graph using Kruskal's algorithm.

Q19. Write a program to find Minimal spanning tree of a graph using Prim's algorithm.

Project Guidelines for UCS 406

1. A group may consist of maximum 2 members.
2. The weightage of the project is of 15 marks.
3. The project must be supported by GUI/Web interface.
4. The project must incorporate suitable data structures and programming methodology (Like: Greedy Approach, Divide and Conquer, Dynamic Programming, Backtracking or Branch and Bound).
5. The use of only Database approach must be avoided until and unless approved by your respective Lab Instructor.
6. Any plagiarism found in the project will result in ***zero*** marks.
7. The Scope and size of the project must be genuine and justified in terms of number of hours to be spent on such work.
8. Project must solve or target some issue or problem to be targeted.
9. New ideas, innovation and new interventions are welcome.
10. The project will be evaluated in 3 phases:
    a. Problem Definition and Problem Solving Approach (During 18th April- 22nd April, 2016).
    b. Working of the designed Algorithmic Solution (During 25th April-29th April, 2016).
    c. The Complete Project (During 2nd May-6th May, 2016).

Question - 1

Maximum difference in an array

Problem Statement

Given an array of integers, a, return the maximum difference of any pair of numbers such that the larger integer in the pair occurs at a higher index (in the array) than the smaller integer. Return -1 if you cannot find a pair that satisfies this condition.

Constraints:
$1 \leq N \leq 1,000,000$
$-1,000,000 \leq a[i] \leq 1,000,000 \ \forall \ i \in [0,N-1]$

Input Format:
The first line of the input is N (the number of elements in the array), and then followed by N elements each in a separate line.

Sample Input 0:
7
2
3
10
2
4
8
1

Sample Output 0:
8

Explanation 0:
For the array { 2,3,10,2,4,8,1} given above, 10 is the largest number in the array and 1 is the smallest number in the array. However, the index of 10 is lower than the lowest index that contains a 1 so the condition of the problem is not satisfied. Using zero-based index notation, the correct answer is a[2] - a[0] = 10 - 2 = 8. This satisfies the condition that the larger number in the pair should be positioned at a higher index in the array than the smaller number.

Sample Input 1:
6
7
9
5
6
3
2

Sample Output 1:

2

Explanation 1:
The value of maxDifference is 9 - 7 = 2.
9 occurs at a[1] and 7 occurs at a[0]. This satisfies the condition that the larger number must have a higher index than the smaller number.

Question - 2
Cut the matchsticks

Problem Statement
You are given N sticks, where each stick is of positive integral length. A cut operation is performed on the sticks such that all of them are reducedby the length of the smallest stick.

Suppose we have 6 sticks of length

```
5 4 4 2 2 8
```

then in one cut operation we make a cut of length 2 from each of the 6 sticks. For next cut operation 4 sticks are left (of non-zero length), whose length are

```
3 2 2 6
```

Above step is repeated till no sticks are left.

Given length of N sticks, print the number of sticks that are cut in subsequent cut operations.

Input Format
You are given an array as function argument which contains N integers: a_0, a_1,...a_{N-1}, separated by spaces, where a_i represents the length of i^{th}stick.

Output Format
For each operation, print the number of sticks that are cut in separate line.

Constraints
$1 \le N \le 1,000$
$1 \le a_i \le 1,000$

Sample Input #1:

```
6
5 4 4 2 2 8
```

Sample Output #1:

```
6
4
2
1
```

Explanation #1:

```
sticks-length       length-of-cut   sticks-cut
```

```
5 4 4 2 2 8         2          6
3 2 2 _ _ 6         2          4
1 _ _ _ _ 4         1          2
_ _ _ _ _ 3         3          1
_ _ _ _ _ _       DONE       DONE
```

Sample Input #2:

```
8
1 2 3 4 3 3 2 1
```

Sample Output #2:

```
8
6
4
1
```

Explanation #2:

```
sticks-length       length-of-cut   sticks-cut
1 2 3 4 3 3 2 1       1             8
_ 1 2 3 2 2 1 _       1             6
_ _ 1 2 1 1 _ _       1             4
_ _ _ 1 _ _ _ _       1             1
_ _ _ _ _ _ _ _     DONE          DONE
```

Question - 1

Poisonous Plants

Problem Statement

There are N plants in a garden. Each of these plants has been added with some amount of pesticide. After each day, if any plant has more pesticide than the plant at its left, being weaker than the left one, it dies. You are given the initial values of the pesticide in each plant. Print the number of days after which no plant dies, i.e. the time after which there are no plants with more pesticide content than the plant to their left.

Input Format

The input consists of an integer N. The next N line consists of N integers describing the array P where P[i] denotes the amount of pesticide in plant i.

Constraints

1 <= N <= 100000

0 <= **P[i]** <= 10^9

Output Format

Output a single value equal to the number of days after which no plants die.

Sample Input

```
7
6
5
8
4
7
10
9
```

Sample Output

```
2
```

Explanation

Initially all plants are alive.
Plants = {(6,1), (5,2), (8,3), (4,4), (7,5), (10,6), (9,7)}
Plants[k] = (i,j) => j^{th} plant has pesticide amount =

i.
After the 1st day, 4 plants remain as plants 3, 5, and 6 die.
Plants = {(6,1), (5,2), (4,4), (9,7)}
After the 2nd day, 3 plants survive as plant 7 dies.
Plants = {(6,1), (5,2), (4,4)}
After the 3rd day, 3 plants survive and no more plants die.
Plants = {(6,1), (5,2), (4,4)}
After the 2nd day the plants stop dying.

Question - 2

Count inversions in an array

Problem Statement

There are N elements in an array. Calculate the number of inversions in an array. `Number of an inversion in array is the number of pair(a[i],a[j]) of elements such that i < j and a[i] > a[j].`

Constraints

$1 \leq N \leq 1{,}000{,}000$
$-1{,}000{,}000 \leq a[i] \leq 1{,}000{,}000 \; \forall \; i \in [0, N-1]$

Input Format:
The first line of the input is N (the number of elements in the array), and then followed by N elements each in a separate line.

Output Format

Output a single value equal to the number of inversions in an array.

Sample Input

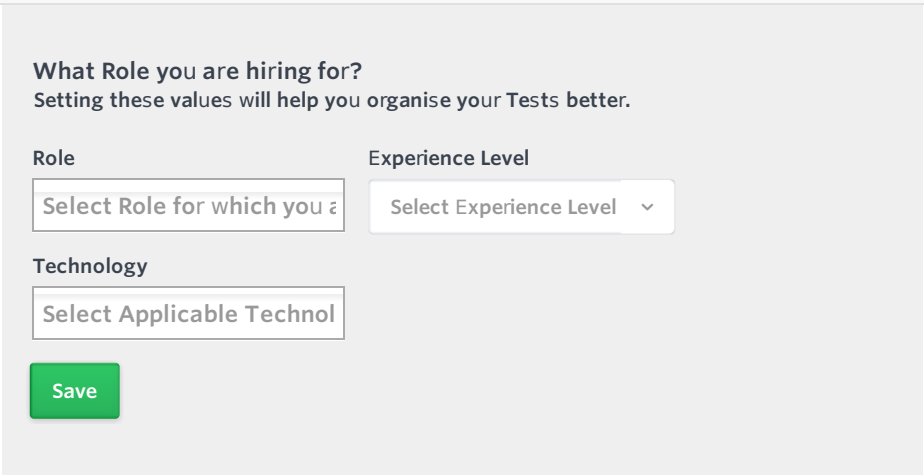
```
5
2
3
6
9
1
```

Sample Output

```
4
```

Explanation

For an example if we have a list of element 2 3 6 9 1 then number of inversion is 4 and the pairs are (2,1), (3,1), (6,1) and (9,1).



Question - 1

REVERSE A SINGLE LINKED LIST

Problem Statement

Write a recursive and non-recursive function to reverse a single-linked list using pointers. Your code should only make a single pass over the list. You can call any function from main for testing your code. If your list is empty, output should be NULL

Input format

To entering data at the end of list, Press 1
Enter a data item to linked list
To stop entering data to linked list, Press 0

SAMPLE INPUT :

1
10
1
20
1
34
0

SAMPLE OUTPUT:

34
20
10

Question - 2

Detecting a cycle in a singly linked list

Problem Statement

Suppose that you are given a single linked list that is either cyclic or acyclic (without a cycle). Write a function that takes as an input

a pointer to the head of a linked list and determines whether the list has cycle or not. If the linked list has cycle then your function should return true, otherwise your function should return false. You can not modify the linked list in any way. **Make your own assumptions while creating a cycle in a linked list.** For example, the linked list below has a cycle.