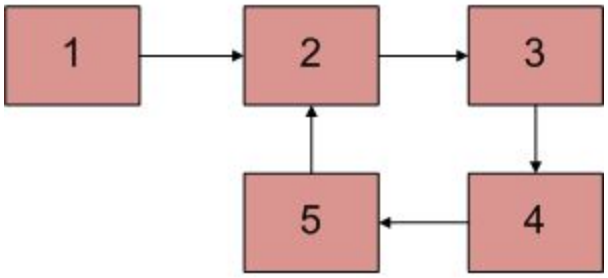
Practice Problems 1 - Recursion & Linked List

1. You are given a collection of n numbers in some order. Give an O(n^2) to find the number of **increasing** subsequences of this word. a **subsequence** is a sequence that can be derived from another sequence by deleting some or no elements without changing the order of the remaining elements. The list of all subsequences for the word "**apple**" would be "*a*", "*ap*", "*al*", "*ae*", "*app*", "*apl*", "*ape*", "*ale*", "*appl*", "*appe*", "*aple*", "*apple*", "*p*", "*pp*", "*pl*", "*pe*", "*ppl*", "*ppe*", "*ple*", "*pple*", "*l*", "*le*", "*e*", "". For simplicity you can assume all numbers are distinct.
2. Suppose you are given an array of integers, having values ranging from 1 to n. Can you give an O(n) algorithm to sort the array? Hint: In general, the fastest time in which you can sort an array is nlogn. Think about what additional information you have here.
3. You are given a pointer to the first node of a linked list of integers. You do not know the number of elements in the list, but you know it is finite. However, you know that there is a cycle in the linked list. Give an algorithm (do not worry about time complexity) to detect the first node of the cycle in the linked list(Assume the first node is not part of the cycle).
   Eg: In this linked list - the node labelled “2” is the first node of the cycle.



4. Given an array A of n distinct elements, we say that an element 1 < i < n is a minima if A[i] < A[i−1] and A[i] <A[i + 1]. The element A[1] is called a minima if A[1] < A[2] and A[n] is the minima if A[n] < A[n−1]. Note that a given array can have multiple minima. Give an algorithm to compute a minima in time O(logn).
5. Consider Euclid's algorithm for finding the gcd of two non-negative numbers n, m.

```
int gcd(n,m) {
        if (m == 0) return n;
        else return gcd(m, n % m);
}
```

   Show that the time required by this algorithm is O(log m).
   Note that it is assumed that each arithmetic operation requires unit cost, irrespective of the size of the number.

Practice Problems 2 - Stacks & Others

6. Given a string of left and right parenthesis, write a function to check that the parenthesis are correctly matched. If they are not, suppose that only one parenthesis (either left or right) has been accidentally missed. Can you identify the position of the missing parenthesis?

7. Suppose you have to transfer the contents of a stack to another stack that is initially empty. All elements of the first stack must appear in the same order in the second and the first stack must become empty. Describe an algorithm for this that uses only constant additional storage (apart from the two stacks). You are only allowed to use the push, pop, top, and empty stack functions. If the stack contains n elements, how many operations are required?

8. a) Write a function which takes a stack of integers as a parameter and returns another stack in which the order of the integers in the stack is reversed. The integer at the top of the original stack is now at the bottom and vice-versa. You can use only the push , pop and empty operations on the stack. What is the time complexity of your algorithm?
   b) Suppose we want to implement the reverse operation as a part of the stack class along with the other operations. Describe an implementation of the stack class so that all operations can be performed in O(1) time.
   c) Specify the reverse operation by writing down its properties.

9. Consider a vector class which has indexing, the usual push back operation and a new pop back operation, which causes the last element to be removed from the vector. Here is a possible implementation. At each point during the execution we will have allocated an array of some M elements. Of these, some $N \leq M$ will be in use, to store the current vector. If we perform a push back, and the required length $N + 1$ becomes bigger than M, we allocate a fresh array of size 2N. We copy the vector elements into the newly allocated array, and delete the old array. If we perform a pop back operation, and if the new vector length $N - 1$ becomes smaller than M/3, then we allocate an array of size $N - 1$, copy the vector into the new array, and delete the old array. And of course we update N, M appropriately.
   Suppose some n operations are performed including creation, push back, pop back and indexing.
   (a) Suppose the rth operation among these is a pop back and causes memory allocation and copying of Q elements. Show that for some constant c, the cQ operations preceding the rth operation do not involve memory allocation.
   (b) Show something similar in case the rth operation is a push back.
   (c) Using the preceding two parts, show that the total work for the n operations must be O(n).

10. Let a_1, a_2, ..., a_n be a sequence of positive integers. Describe an O(n) time algorithm to find a subsequence with maximum sum such that no three consecutive elements of the initial sequence are contained in the subsequence.

11. Given a sequence of n bits, that is 0's and 1's, find the length of the longest substring containing equal number of 0's and 1's in O(n) time.

12. Let a_0, a_1, ..., a_{n-1} be a sequence of integers. Find for each i, the smallest j such that j > i and a_j < a_i, if such a j exists. This should be done in O(n) time.