GeeksMan

Data Structure

Lesson 4

Introduction to Stack Of Pair STL

Trilok Kaushik

Founder of GeeksMan

Chirag Soni

Nupur Pahuja

Jessica Mishra

Team Coordinators

Stack of Pair in C++ STL

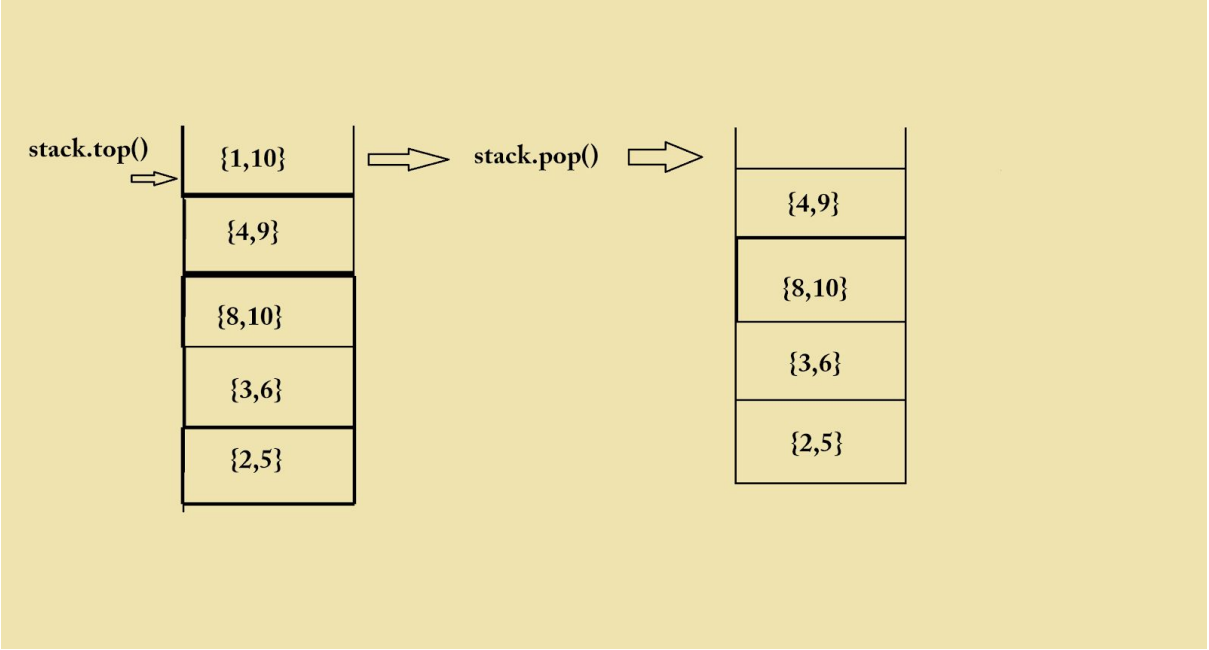
Stack in STL : Stacks are a type of *container adaptors* with LIFO(Last In First Out) type of working, where a new element is added at one end and (top) an element is removed from that end only.

Pair In STL : The pair container is a simple container , consisting of two data elements or objects. The first element is referenced as 'first' and the second element as 'second' and the order is fixed (first, second).

Syntax :

```
stack < pair < datatype , datatype > > stack_of_pair;
```

Example :



EXAMPLE :

```cpp
#include <bits/stdc++.h>
#include <iostream>
using namespace std;

int main()
{
    stack<pair<char , int >>st;
    // pushing pair of elements in the stack
       st.push({'a',1});
      st.push({'b',2});
      st.push({'c',3});
      st.push({'d',4});
      st.push({'e',5});

   //print all the pairs of the stack
      for(int i = 0 ;i < 5;i++)
      {
          cout << i+1 << "th pair = (" << st.top().first
  << " , " << st.top().second << ")"  <<endl;
          st.pop();
      }
     return 0;
  }
```

OUTPUT :

stdout

```
1th pair = (e , 5)
2th pair = (d , 4)
3th pair = (c , 3)
4th pair = (b , 2)
5th pair = (a , 1)
```

Print Bracket Number

Given an expression **exp** of length **n** consisting of some brackets. The task is to print the bracket numbers when the expression is being parsed.

Example :

Input : (a+(b*c))+(d/e)

Output : 1 2 2 1 3 3

Algorithm :

1. Make a stack of pair with the first element as character dtype and second as integer.
2. Traverse the string from left to right.
3. Initialize a count variable with 0.
4. While traversing the string , if we come across an opening bracket ' (' , then push this character along with ++count in the stack and print count .
5. If we come across a closing bracket ') ' , then print the second element of the top of the stack and then pop it.

Code :

```
1. #include <bits/stdc++.h>
2. #include <iostream>
3. using namespace std;
4.
5. void myfun()
6. {
```

```
    string s;
    cin >> s;
    int l = s.length();
       stack<pair<char,int>>st;
       int count = 0;
       for(int i=0;i<l;i++)
       {
           if(s[i] == '(')
           {
               count++;
               st.push({s[i],count});
               cout << count << " ";
           }
           else if(s[i] == ')')
           {
               cout << st.top().second << " ";
               st.pop();


           }
       }
       cout << endl;
   }
   int main()
   {
       int t;
       cin >> t;
       while(t>0)
       {
           myfun();
           t--;
       }
   }
```

Link of the code : https://sapphireengine.com/@/iccwzo

The Celebrity Problem

You are in a party of **N** people, where only one person is known to everyone. Such a person **may be present** in the party, if yes, **(s)he doesn't know anyone** in the party. Your task is to find the stranger (celebrity) at the party.

```
Input:
N = 3
M[][] = {{0 1 0},
         {0 0 0},
         {0 1 0}}
Output: 1
Explanation: The matrix will look like
0 1 0
0 0 0
0 1 0
Here,  the celebrity is the person with
index 1 ie id 1
```

```
Input:
N = 2
M[][] = {{0 1},
         {1 0}}
Output: -1
Explanation: The matrix will look like
0 1
1 0
Here, there is no such person who is a
celebrity (a celebrity should know no
one).
```

SOLUTION - 1

Expected Time Complexity: O(N)
Expected Auxiliary Space: O(N)

ALGORITHM

1. Create a stack and push the number of rows in it (Matrix is of size n*n).
2. Let a = stack.top() ,
   stack.pop() and b = stack.top() ,
   stack.pop().
3. If a knows b , then push b in stack , else push a.
4. Repeat step 2 and 3 until stack.size() == 1.
5. Check for each person , if stack.top() knows anyone , there is no celebrity ,
   Else if anyone in the party doesn't know stack.top() , there is no celebrity ,
   Else stack.top() is the celebrity.

CODE:

```
1. #include<bits/stdc++.h>
2. using namespace std;
3.
4. #define MAX 501
5.
6. int getId(int M[MAX][MAX],int n);
7.
8. int main()
9. {
10.     int T;
11.     cin>>T;
12.     int M[MAX][MAX];
13.
14.     while(T--)
15.     {
```

```
16.          int N;
17.          cin>>N;
18.
19.          memset(M,0,sizeof M);
20.
21.          for(int i=0;i<N;i++)
22.          {
23.              for(int j=0;j<N;j++)
24.              {
25.                  cin>>M[i][j];
26.              }
27.          }
28.          cout<<getId(M,N)<<endl;
29.      }
30.  }
31.
32.  // } Driver Code Ends
33.
34.
35.  // The task is to complete this function
36.
37.  // M[][]: input matrix
38.  // n: size of matrix (n*n)
39.
40.  int getId(int M[MAX][MAX], int n)
41.  {
42.      //Your code here
43.
44.      stack<int> st;
45.      int i,a,b;
46.
47.      for(i=0;i<n;i++)
48.          st.push(i);
49.
50.      while(st.size()!=1)
51.      {
52.          a=st.top();
```

```
            st.pop();
            b=st.top();
            st.pop();
            if(M[a][b]==1)
                st.push(b);

            else
            st.push(a);

        }
        int c=st.top();

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

            if(i==c)
            Continue;

            if(M[i][c]==0)
            {
                return -1;
            }

            if(M[c][i]==1)
            {
                return -1;
            }

        }
        return c;
    }
```

Link for the above code : https://sapphireengine.com/@/fs6zsj

SOLUTION - 2

Expected Time Complexity: O(N)
Expected Auxiliary Space: O(1)

ALGORITHM

1. Define two variables , a and b which points to the index of the row , and column respectively.Take a = 0, and b = n-1 (Size of matrix = n*n).
2. If a knows b , then do a++ else do b-- .
3. Repeat step 3 while(a<b).
4. Check for each person , if someone doesnt know a or if a knows someone , then , there is no celebrity,
   Else a is the celebrity,.

CODE :

```
#include<bits/stdc++.h>
using namespace std;
#define MAX 501

int getId(int M[MAX][MAX],int n);

int main()
{
  int T;
    cin>>T;
    int M[MAX][MAX];
    while(T--)
    {
        int N;
        cin>>N;
        memset(M,0,sizeof M);
        for(int i=0;i<N;i++)
        {
            for(int j=0;j<N;j++)
            {
                cin>>M[i][j];
```

```
            }
        }
        cout<<getId(M,N)<<endl;

    }
}

bool knows(int M[MAX][MAX],int a, int b)
{
   return M[a][b];
}
int getId(int M[MAX][MAX], int n)
{
   int a = 0;
   int b = n - 1;
   while (a < b)
   {
        if (knows(M,a, b))
             a++;
        else
             b--;
   }
   for (int i = 0; i < n; i++)
   {
        if ( (i != a) &&
                  (knows(M,a, i) ||
                  !knows(M,i, a)) )
             return -1;
   }
   return a;
}
```

Link for the Code : https://sapphireengine.com/@/pvhdx0

Decode the string

An encoded string (s) is given, the task is to decode it. The pattern in which the strings were encoded were as follows

Input : 1[b] **Output** : b

Input : 3[b2[ca]] **Output** : bcacabcacabcaca

ALGORITHM :

1. Create two stacks , one to store integral values , and the other one to store the characters / alphabets.
2. Traverse the string from left to right.
3. Whenever we encounter any number, push it into the integer stack and in case of any alphabet (a to z) or open bracket ' [', push it onto the character stack.
4. Whenever any close bracket '] ' is encountered , pop the character from the character stack until an open bracket ' [' is not found in the character stack. Also, pop the top element from the integer stack , say n. Now make a string repeating the popped character n times. Now, push all characters of the string in the stack.

CODE :

```
1. #include<bits/stdc++.h>
2. #include <iostream>
3. using namespace std;
4.
5. string myfun(string s)
6. {
```

```
7.    stack<int>it;
8.    stack<char>ch;
9.    string temp = "", result = "";
10.      for(int i = 0; i < s.length() ; i++)
11.      {
12.          int count = 0 ;
13.          if(s[i] >= '0' && s[i] <='9')
14.          {
15.              while(s[i] >= '0' && s[i] <= '9')
16.              {
17.                  count = count*10 + s[i] - '0';
18.                  i++;
19.              }
20.              i--;
21.              it.push(count);
22.          }
23.          else if(s[i] == '[')
24.          {
25.              if(s[i-1] >= '0' && s[i-1] <='9')
26.                  ch.push(s[i]);
27.          }
28.          else if(s[i] == ']')
29.          {
30.              temp = "";
31.              count = 0;
32.
33.              if (!it.empty())
34.              {
35.                  count = it.top();
36.                  it.pop();
37.              }
38.
39.              while (!ch.empty() && ch.top()!='[' )
40.              {
41.                  temp = ch.top() + temp;
42.                  ch.pop();
43.              }
```

```
44.
45.                 if (!ch.empty() && ch.top() == '[')
46.                     ch.pop();
47.
48.                 for (int j = 0; j < count; j++)
49.                     result = result + temp;
50.                 for (int j = 0; j < result.length(); j++)
51.                     ch.push(result[j]);
52.
53.                 result = "";
54.             }
55.             else
56.                 ch.push(s[i]);
57.         }
58.          while (!ch.empty())
59.         {
60.             result =ch.top() + result;
61.             ch.pop();
62.         }
63.         return result;
64.     }
65.     int main()
66.     {
67.         int t;
68.         cin >> t;
69.         while(t--)
70.         {
71.             string st;
72.             string s;
73.             cin >> s;
74.             st = myfun(s);
75.             cout << st<< endl;
76.         }
77.        return 0;
78.     }
79.
```

Link for the code: https://sapphireengine.com/@/x4y6vp

1. https://practice.geeksforgeeks.org/problems/string-manipulation/0
2. https://practice.geeksforgeeks.org/problems/evaluation-of-postfix-expression/0
3. https://practice.geeksforgeeks.org/problems/count-the-reversals/0
4. https://practice.geeksforgeeks.org/problems/easy-string/0
5. https://practice.geeksforgeeks.org/problems/maximum-difference-1587115620/1