STANDARD TEMPLATE LIBRARY

The problem

Problems which involve usage of standard data structure becomes

- Difficult to solve
- Time-consuming
- Losing focus on main problem statement

in C as we have to implement data structure first.

Designing a Library containing different standard Data Structures and Algorithms

Step 1

Design fast and efficient implementation of

- Linked List
- Stacks
- Queues
- Binary Trees

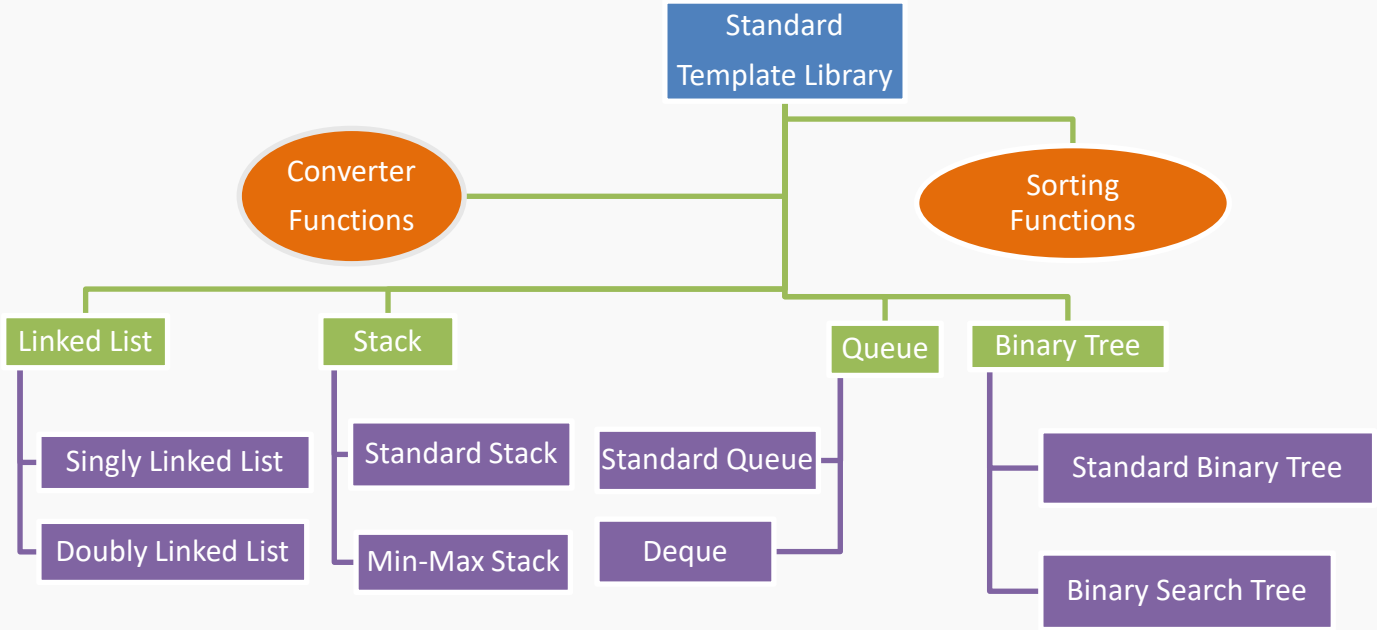
Step 2

Implementing the structures and there sub-structures(MinStack, Deque, BST, etc) and their functions for various data-types

Step 3

Linking all the structures together in a file

How it works


Standard Template Library
Converter Functions
Sorting Functions
Linked List
Stack
Queue
Binary Tree
Singly Linked List
Doubly Linked List
Standard Stack
Min-Max Stack
Standard Queue
Deque
Standard Binary Tree
Binary Search Tree

DEMO

```c
#include <stdio.h>
#include "stlclib.h"

int main(){
        int n1=1,n2=2,n3=3,n4=4,n5=5;
        int arr[5]={3,4,1,5,2};
        float f[5]={3.414,0.532,1,2.23,1.55};
        float f1[5]={1,1.414,1.732,2,2.24};
        char c[5]={'d','f','a','i','c'};
        char* s[5]={"Harsh","arsh","Sandhu","Arsh","sandhu"};

        char* check="aarsh";
        float ff=1.55;
        printf("\n\nSearching Check\n\n");
        printf("%d\n",searchArr(s,5,&check,4));
        printf("%d\n",searchArr(f,5,&ff,3));

        char c1='a',c2='b',c3='c',c4='d',c5='e';
        printf("\n\nCharacter Linked List\n\n");
        linkedlist* node1=constructLinkedList(2);
        deleteAtHead(node1);
        insertAtTail(node1,&c1);
        insertAtTail(node1,&c2);
        insertAtTail(node1,&c3);
        insertAtHead(node1,&c4);
        insertAtHead(node1,&c5);
        deleteAtTail(node1);
        deleteAtHead(node1);
        printLinkedlist(node1);

        printf("\n\nInteger Linked List\n\n");
        linkedlist* node2=constructLinkedList(1);
        int arr1[5]={1,2,3,4,5};
        void* ptr[5];
        itovp(arr1,ptr,5);
        insertArrAtTail(node2,ptr,5);
        printLinkedlist(node2);


        linkedlist* node3=constructLinkedList(3);

        printf("\n\nFloat Linked List\n\n");
        float arr2[5]={1,1.414,1.732,4,2.24};
        void* ptr2[5];
        ftovp(arr2,ptr2,5);
        insertArrAtHead(node3,ptr2,5);

        printLinkedlist(node3);

        printf("\n\n Check cycle -1\n\n");
        printf("%d\n",detectCycle(node1));
        makeCycle(node1,3);
        printf("%d\n",detectCycle(node1));
        removeCycle(node1);
        printLinkedlist(node1);
        printf("%d\n",detectCycle(node1));

        printf("\n\n Check cycle -2\n\n");
        printf("%d\n",detectCycle(node2));
        makeCycle(node2,2);
        printf("%d\n",detectCycle(node2));
        removeCycle(node2);
        printLinkedlist(node2);
        printf("%d\n",detectCycle(node2));


        printf("\n\n DEQUE \n\n");
        deque* q=constructDeque(1);
        enqueueBackArr(q,ptr,5);
        while(!is_emptyDeque(q)){
                printf("%d %d\n",*(int*)peekFront(q),*(int*)peekBack(q));
                dequeueFront(q);
        }
        printf("\n");
}
```

```
vagrant@myvm18:~/STL_LIB$ vi test.c
vagrant@myvm18:~/STL_LIB$ gcc test.c -L. -lcstl
vagrant@myvm18:~/STL_LIB$ ./a.out


Searching Check

-1
4


Character Linked List

ERROR: Linked-list is empty
d a b


Integer Linked List

1 2 3 4 5


Float Linked List

2.240000 4.000000 1.732000 1.414000 1.000000


 Check cycle -1

0
ERROR: Cant Create Cycle
0
ERROR: No Cycle to remove
d a b
0


 Check cycle -2

0
1
1 2 3 4 5
0


 DEQUE

1 5
2 5
3 5
4 5
5 5
```

STANDARD TEMPLATE LIBRARY

Created by: Harsh Sandhu
harshsandhu.cse19@chitkarauniversity.edu.in

GitHub link:- https://git.io/JuXN6

ReadMe:- https://git.io/JuXN7