26/12/2020

# (11) Text Editor (Stack)

Our main focus in this project will be to implement UNDO functionality used in the text editors using stack ds. When UNDO button'll be pressed last deleted or added letters will reapear or delete.
get.

and on pressing clear box all the letters will get removed. Uptill now we won't be able to edit our text from between!
or UNDO

→ challenge

Manipulate or update the code to do so!

* logics & concepts

- Stack
- UNDO operation
- Buffer concept
- Preventing user to copy/past & the data/text (with either ctrl+c/p or via mouse)

# Stack Implementation (we've used arrays ie why we can easily implement `LL` and reduce tc to O(1)).

so that we could use this file or import it in another file
`export {stack}`

```
class Stack {
    constructor(){
        this.size = 0;
        this.buffer = 4;
        this.stack = [];
    }
    clear(){
        this.size = 0;
        this.stack = [];
    }
    isEmpty(){
        return (this.size === 0);
    }
    top(){
        return this.stack[this.size-1];
    }
    pop(){
        if (!this.isEmpty()){
            this.size--;
            return this.stack.pop();
        }else{
            return[-1,''];
        }
    }
    push(type, char){
        if(this.isEmpty()){
            if (type===0)
                this.stack.push([type,1]);
        }else{
            let tmp = this.top();
            if(tmp[0]===type &&
               tmp[1].length < this.buffer){
                let top = this.pop();
                top[1] = char + top[1];
                this.stack.push(top);
            }else{
                this.stack.push([type,char]);
            }
            this.size++;
        }
    }
}
```

mtlb ek baar me stack ke ek elem me itne chars honge.

Basically it colects at max 4 consecutive operaths of same type. It means that, supose we input 'abcdef' (all these are insert oprtns ie type=0) so wht it does that in 1st elem it'll store abcd, and ef in othr for separate oprtns. Supose we press UNDO so first 'f' will get erased the abcd;

whn stack is empty you can perform only insert operath.

to chck if top elem ka type rec- ntly per- fmd oprtn ke equal hai kya if yes then push.

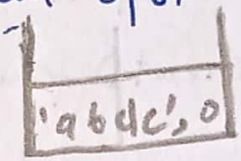
type=0 → Insert
type=1 → delt

basically in 3 cases we put a new element at top of the stack:

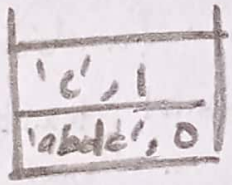
(st) (i) whn st is empty.

(ii) similar operatns are performed & size of top = Buffer

(iii) Different oprtn is performed

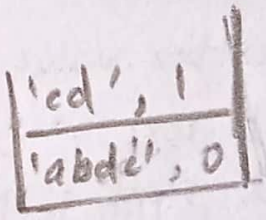
ex - 'abdc' were input. (inserted) → 


- then user delted 'c'
since a new oprtn is performed → we'll create a new elem & put it at top of stack



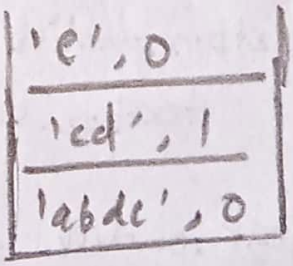

op
abd

- then user again use delt oprtn
as type is sm & top elem size is less than bufr. → we'll add 'd' in top



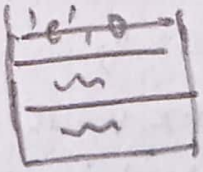

op
ab

- then user inserted 'e'
as its diff oprtn or of diff type new elem is inserted at top →



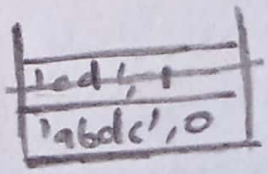

op
abc

- now if we press UNDO (top elem will get cleard) →




op
ab

- if UNDO is again pxd (as we r kept record op oprtns we delted we'll retain in op) →




op
abdc

# Script.js

```
import {stack} from "./stack.js";

document.onkeydown = function (event){
    if (event.ctrlKey || event.metaKey){
        event.preventDefault();
    }
};
```

Disable copy & paste fns by disable usage of ctrl.

mtlb jb pora pg load ho jaye then run this fn.

```
onload = function(){
    const textbox = document.getElementById...   → where we type
    "  undo ...
    "  clear ...
    "  temptext ...                              → "what oprat" is performd" is displayed here

    textbox.value = "";
    let text = "";
    let stack = new stack();
```

through this we ensures that the cursor does not any where! it only remain in start or end

(ex) abc| → if cursor is here

a|bc → we can't bring it over here

```
    textbox.onclick = function(){
        textbox.selectionStart = textbox.selectionEnd = textbox.value.length;
    }

    clear.onclick = function(){
        stack.clear();
        text = "";
        textbox.value = "";
        temptext.innerHTML = "Sequence of oprt'n will be shown here";
    };
```

to clear all

```
textbox.oninput = function (event) {
```
→ it's this gives information about the operation like insert, delete etc

when a user does any operation in textbox this fn'll be called

```
    switch (event.inputType) {
        case "insertText":
            stack.push (0, event.data);
            break;
```
} if any alphabet is pressed this'll get executed

if backspace key is prsd this'll get executed

```
        case "deleteContentBackward":
            Stack.push (1, text[text.length - 1]);
            break;
    }
```

whtevr oprtn is performed, it gets display in this space

```
    temptext.innerHTML = "In Stack" + stack.top() +
                         "<br>" + temptext.innerHTML;
    text = textbox.value;
};
```
→ stores the value which is in the text box.

```
undo.onclick = function () {
    let operation = stack.pop();
    if (operation[0] !== -1) {
        temptext.innerHTML = "Performing UNDO opr <br>" + ;
        if (operation[0] === 1) {
            let len = operation[1].length;
            textbox.value = textbox.value.substring (0, textbox.value.length - len);
        } else {
            textbox.value += operation[2];
        }
        text = textbox.value;
    };
};
```
temptext.innerHTML

(0, 6-4) = (0,2)

if top elem or last performd oprtn was delt then this line add last operated cmds string

to keep the track of the textbox (8)

if we had 'gfabcd' in our textbox (ie len = 6). So wht this line does is it select all chars from 0 till 2 (excluded) ie (g f) ∴ removing "abcd".

0 1 2 3 4

0 1