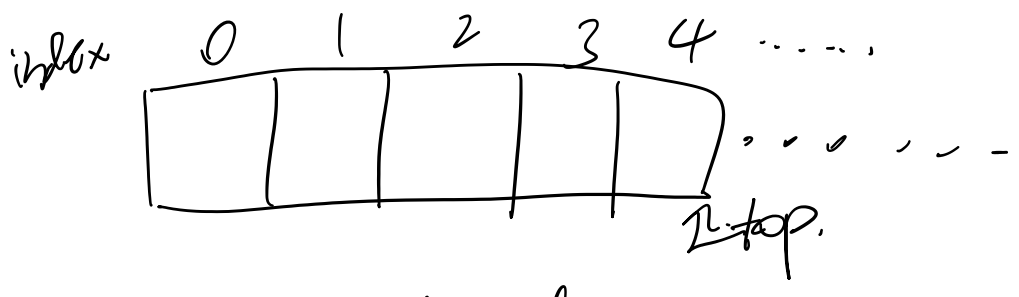
Stack implemented by array

Saturday, 11 February 2023 21:37



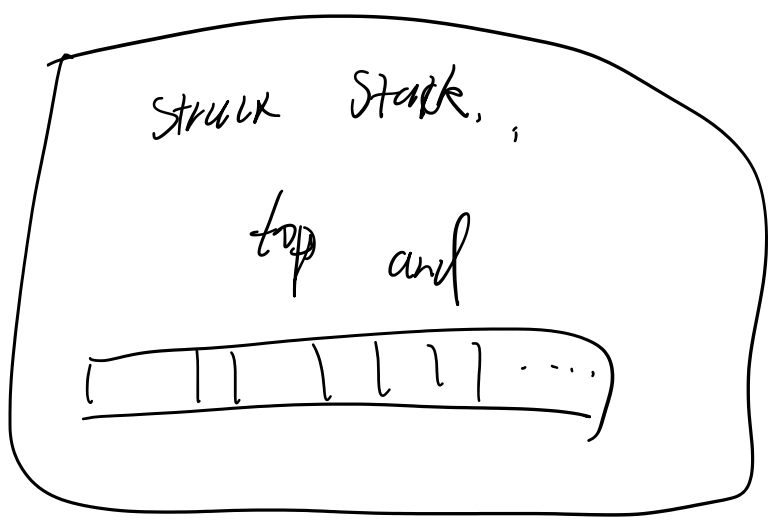
top is the index of array.

$0 \le top \le$ size of Array -1

top $\in [0,$ size $-1]$

```
typedef int DATA_TYPE;

#define SIZE 5
typedef struct Stack{
    int top; //index
    DATA_TYPE stack[SIZE];
} Stack;
```



functions:

① initialize( )
② push( )
③ pop( )
④ peek( )
⑤ displayInt( )

① initialize(Stack* stack)

stack -> top = -1

② push(Stack* stack, DATA_TYPE value)

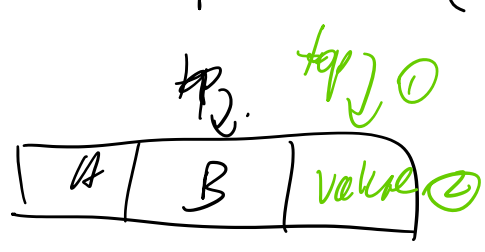
Special case: stack is full. i.e. (top = size - 1)

return;

General case:

top++;

stack.stack[top] = value;

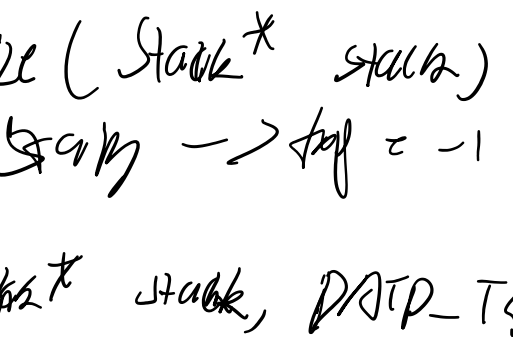


③ pop(Stack* stack)

Special case: Stack is empty, i.e. top = -1

return -1;

General case:



return_value = stack.stack[top];

top--;

④ peek(Stack* stack)

Special case: empty, i.e. top = -1

return;

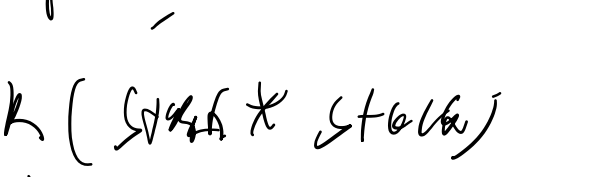
General case: top

return stack.stack[top];

⑤ displayInt(Stack* stack)

Special case: empty ...

General case:



loop from top to 0.

for(int i = top; i >= 0; i--)

.....