Day 8 assignment

Ans1.

Ans2.

Implement push, pop and find the minimum element in a stack in O(1) time complexity.

struct stack {

int top;

int capacity;

int \*arr;

};

int min = 0;

////////////////////////// create stack

struct stack\* createstack(int cap){

struct stack \*newstack = (struct stack\*)malloc(sizeof(struct stack));

newstack->capacity = cap;

newstack->top = -1;

newstack->arr = (int\*) malloc(newstack->capacity \* sizeof(int));

return newstack;

}

//// check weather stack is full or not

int isfull(struct stack \*s){

if(s->top == s->capacity-1){

return 1;

}else{

return 0;

}

}

//// check weather stack is empty or not

int isempty(struct stack \*s){

if(s->top == -1){

return 1;

}else{

return 0;

}

}

//// push into stack

void push(struct stack \*s,int ele){

if(!isfull(s)){

if(ele >= min){

min = ele;

}

s->top++;

s->arr[s->top] = ele;

printf("element inserted \n");

}else{

printf("Stack is full \n");

}

}

//// pop into stack

void pop(struct stack \*s){

if(!isempty(s)){

int temp = s->arr[s->top];

s->top--;

printf("Element is pop %d",temp);

}else{

printf("Stack is empty \n");

}

}

//// get min element in stack O(n)

void getmin(struct stack \*s){

if(!isempty(s)){

printf("Min Element is: %d",min);

}else{

printf("Stack is empty \n");

}

}

main(){

struct stack \*s = createstack(2);

pop(s);

push(s,10);

push(s,20);

push(s,30);

getmin(s);

}