Apex College

BCIS Program

Affiliated to Pokhara University

Activity ?

Checked
4.5.22

Data Structure & Algorithms
Lab Report

1. stack operations : PUSH, POP

Date: 04-05-2022

Submitted by:
Ishwor Shrestha
Roll no.: 2018-BCIS-414

Submitted to:
Pravakar Ghimire, &
Anmol Shrestha
Apex College

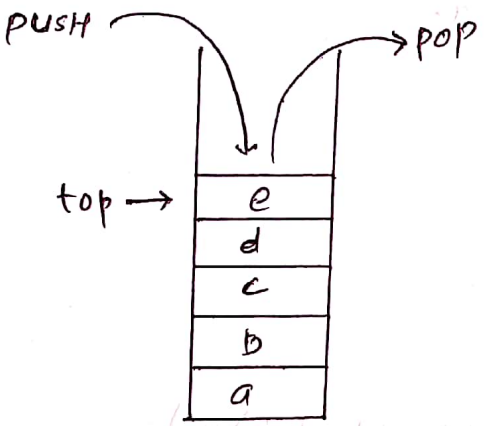
Lab 1 Objectives

- To understand the stack and it's operations
- To create and execute algorithm and program using PUSH and POP operations.

Introduction

Stack is a linear data structure which stores data temporarily to perform PUSH & POP operations in LIFO sequence.

The concept, top of the stack pointer is used to insert items or delete items based on LIFO sequence,



A menu driven program to demonstrate the operations in the stack.

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 10

struct stack {
  int items[MAX];
  int top;
};
typedef struct stack stack;
```

```
void push(stack *s, int elt){
    if(s-> top == MAX-1)
      printf("stack is full. \n");

    else {
      s->top++;
      s->items[s->top] = elt;
    }
}

int pop(stack *s) {
    int elt = -1;
    if (s->top == -1)
      printf("stack is empty. \n");

    else {
      elt = s->items[s->top];
      s-> top--;
    }
    return elt;
}

void display (stack *s) {
    int i;
    printf("Elements (items) in stack: \n");
    for(i= s->top; i>=0; i--){
      printf("%d \n", s-> items[i]);
    }
}

void main() {
    stack s;
    int ch, item;
    s. top = -1;
```

```
while (1) {
    printf("Enter a number from following choices : \n");
    printf("1. PUSH \n 2. POP \n 3. DISPLAY \n 4. EXIT \n");
    scanf("%d", &ch);

    switch(ch) {

        case 1:
            printf("Enter data to PUSH : ");
            scanf("%d", &item);
            push(&s, item);
            break;

        case 2:
            item = pop(&s);
            printf("%d is popped. \n", item);
            break;

        case 3:
            display(&s);
            break;

        case 4:
            exit(0);

        default:
            printf("Invalid choice. \n");

    }
}
```

#Activities

- Created 4 different menu items to perform following activities

① PUSH

- push operation is used to insert or add new elements in the top of the stack.
  - Check whether the stack is full or not.
    - `if(s->top == MAX-1)`
      `print "stack is full" & exit`
  - Otherwise, stack is not full
    - increment: value of the top
      `s->top++;`
    - insert value at the top
      `s->items[s->top] = elt.`

② POP

- Pop operation is used to remove or delete the element from the top of the stack.
  - Check whether the stack is ~~empty~~ empty or not.
    - `if(s->top == -1)`
      `print "stack is empty" & exit`
  - Otherwise, stack is not empty
    - set, `elt = s->items[s->top];`
    - decrement: value of top
      `s->top--;`
    - return elt;

③ DISPLAY

- Display operation is done for displaying or traversing all the elements available in the stack
  - print the value pointed by top
  - decrement: value of ponter
  - Keep performing activities of above until you get all elements.

④ EXIT
- Exit is used to terminate the whole program.
  - exit(0);

#Conclusion

I have learned about the different operations performed in stack i.e. push & pop. With concept, We get knowledge to actual implementation ~~using pr~~ in program.