백준 | 10828번 스택

통계 수정 삭제

khi8930 · 32분 전

0

백준 스터디

2023 Algorithm

목록 보기 1/2

- 소요시간 : 2시간 이상
- 자료 참고 여부 : X (기존에 공부한 자료만 참고)
- 체감 난이도 : 中

📌 10828번 문제

10828번 제출 맞힌 사람 숏코딩 재채점 결과 채점 현황 내 제출 난이도 기여 강의 질문 게시판

스택 성공

실버 IV

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
0.5 초 (추가 시간 없음)	256 MB	231211	82649	60125	37.235%

문제

정수를 저장하는 스택을 구현한 다음, 입력으로 주어지는 명령을 처리하는 프로그램을 작성하시오.

명령은 총 다섯 가지이다.

- push X: 정수 X를 스택에 넣는 연산이다.
- pop: 스택에서 가장 위에 있는 정수를 빼고, 그 수를 출력한다. 만약 스택에 들어있는 정수가 없는 경우에는 -1을 출력한다.
- size: 스택에 들어있는 정수의 개수를 출력한다.
- empty: 스택이 비어있으면 1, 아니면 0을 출력한다.
- top: 스택의 가장 위에 있는 정수를 출력한다. 만약 스택에 들어있는 정수가 없는 경우에는 -1을 출력한다.

- https://www.acmicpc.net/problem/10828

1. 죽일놈의 엔터키
2. 문제의 조건을 잘보자 .. 최대 입력값의 범위
3. **strcmp** .. 잘 기억하자 ! (단어는 문자가 아님, 문장으로 봐야함)

코드 전문

```
#pragma warning(disable:4996)
#pragma warning(disable:4013)
#include<stdio.h>
#include<string.h>
#define MAX_NUM 1000000
```

```c
typedef struct number {
    int NUM;
}num;

typedef num Element;
Element data[MAX_NUM];
int top;

void error(char str[]) {
    printf("%s", str);
}

void init_stack() {
    top = -1;
}

int is_empty() {
    return top == -1;
}

int is_full() {
    return top == MAX_NUM - 1;
}

int size() {
    if (top <= -1) {
        return 0;
    }
    else {
        return top + 1;
    }

}

void push(Element s) {

    data[++top] = s;
}

Element pop() {

    if (top <= -1) {
        top = 0;
        printf("-1\n");

    }
    else {
        printf("%d\n", data[top]);
    }
    return data[top--];

}

Element peek() {
    if (is_empty()) {
        printf("-1\n");
    }
    else {
        printf("%d\n", data[top]);
    }

    return data[top];
}

num get_num(int number) {
    num N;
    N.NUM = number;

    return N;
}

int main() { // 러닝타임 에러 .. why ?
    int scale, n;
```

```
int i = 0;
char a[100];

init_stack();

scanf("%d", &scale);

for (i = 0; i < scale; i++) {


    scanf("%s", &a);
    getchar();

    if (strcmp(a, "push") == 0) { // strcmp !!!
        scanf("%d", &n);
        push(get_num(n));
    }
    else if (strcmp(a, "pop") == 0) {
        pop();
    }
    else if (strcmp(a, "size") == 0) {
        printf("%d\n", size());
    }
    else if (strcmp(a, "empty") == 0) {
        printf("%d\n", is_empty());
        }
        else if (strcmp(a, "top") == 0) {
            peek();
        }


    }

    return 0;

}
```

코드 해설

```
#define MAX_NUM 1000000

typedef struct number {
    int NUM;
}num;

typedef num Element;
Element data[MAX_NUM];
int top;
```

- 문제 조건에 맞춰 스택의 최대 길이는 10000 이상으로 설정
- 스택에 들어가는 정수는 struct문을 써서 입력
- 스택에 들어가있는 요소의 수를 알기 위해 전역변수 top 사용

```
void init_stack() { // 초기화 함수
    top = -1;
}

int is_empty() { // empty 확인
    return top == -1;
}


int size() { // 스택의 사이즈 확인
    if (top <= -1) {
        return 0;
    }
    else {
```

```
        return top + 1;
    }

}

void push(Element s) {

    data[++top] = s;
}

Element pop() {

    if (top <= -1) {
        top = 0;
        printf("-1\n");

    }
    else {
        printf("%d\n", data[top]);
    }

    return data[top--];

}
```

```
Element peek() { // 가장 위에 있는 숫자만
    if (is_empty()) {
        printf("-1\n");
    }
    else {
        printf("%d\n", data[top]);
    }

    return data[top];
}

num get_num(int number) {
    num N;
    N.NUM = number;

    return N;
}
```

- 스택이 비어있는지 판단하기 위해서 top의 개수를 이용함
    - 이후에 is_empty() 함수를 활용하여 스택이 비어있는지 확인한 버전도 있었는데 어디간지 모르겠음.
- 이때 사용한 push(), pop() 함수는 작년 자료구조 수업에서 사용한 함수들을 참고
- get_num() 함수는 구조체에 입력받은 변수를 넣어주기 위해서 사용
    - 근데 별로 중요한 건 아닌 듯 하다. 구조체가 변수 하나기 때문에 다른 형식으로 표현하거나 아예 없앨 수도 있을 것 같다.

```
int main() { // 러닝타임 에러 발생 전적 多
    int scale, n;
    int i = 0;
    char a[100];

    init_stack();

    scanf("%d", &scale);

    for (i = 0; i < scale; i++) {


        scanf("%s", &a);
        getchar();

        if (strcmp(a, "push") == 0) { // strcmp !!!
            scanf("%d", &n);
            push(get_num(n));
```

```
        }
        else if (strcmp(a, "pop") == 0) {
            pop();
        }
        else if (strcmp(a, "size") == 0) {
            printf("%d\n", size());
        }
        else if (strcmp(a, "empty") == 0) {
            printf("%d\n", is_empty());
        }
        else if (strcmp(a, "top") == 0) {
            peek();
        }


    }

    return 0;

}
```

- 스택, 큐, 트리 같은 자료구조는 main 함수가 복잡하진 않다.
- 변수 'a'의 러닝타임 에러가 몇번이나 났는데 이유를 찾지 못하고 어느샌가 해결되었다.
- 2시간을 쓰게 만든 주범 : 출력 양식에서 ₩n 사용 안했다고 출력예시가 안맞다고 나옴 .. ^^

📌 총평

문제 자체의 난이도는 그닥 높지 않은 것 같음

그러나 문제를 제대로 풀기 위해서는 스택에 대한 제대로 된 이해가 필요하단 점에서 시간이 많이 걸렸음. (오랜만에 C도 보고 문제도 품.)

엔터키 이슈와 연이은 '틀린 문제' 선언으로 인해 멘탈이 좀 나간거 빼고는 다 괜찮았다.

Hyangim

코딩 공부 중인 대학생 👩‍💻

다음 포스트

백준 | 10845번 큐

백준 | 10845번 큐

통계 수정 삭제

khi8930 · 방금 전

0

백준 스터디

2023 Algorithm

목록 보기 2/2

- 소요시간 : 30분 ~ 1시간
- 자료 참고 여부 : O (기존에 공부하던 자료가 틀린 자료였음)
- 체감 난이도 : 中

📌 10845번 문제

큐 성공

실버 IV

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
0.5 초 (추가 시간 없음)	256 MB	111515	51541	40479	48.970%

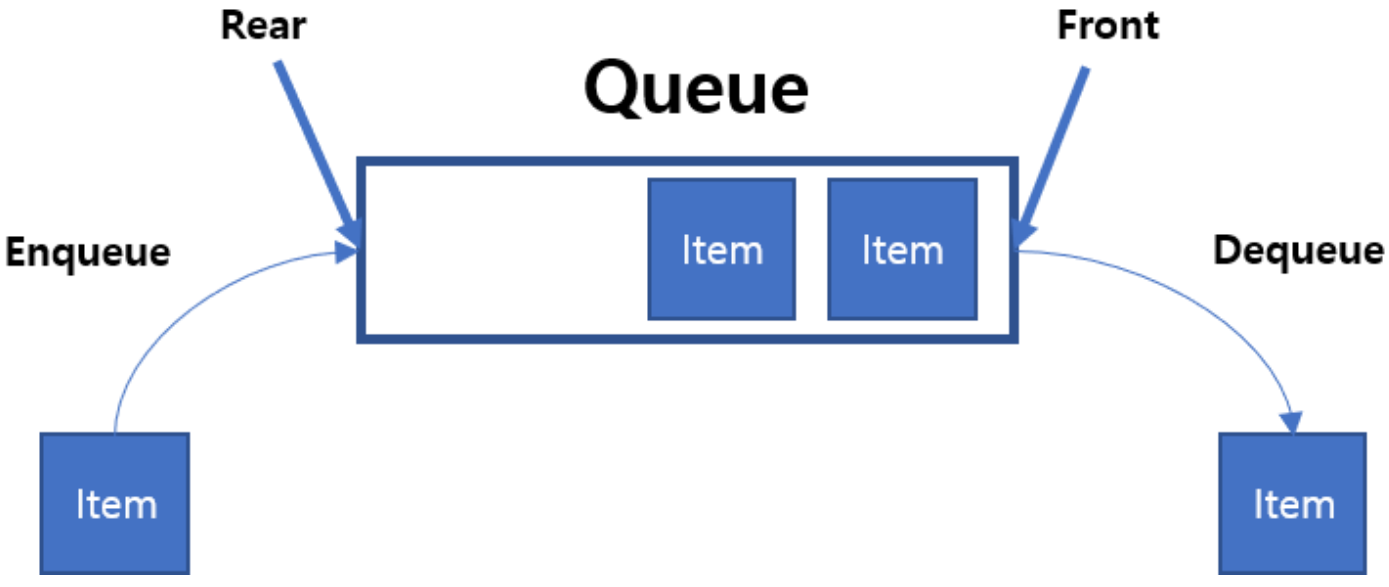
문제

정수를 저장하는 큐를 구현한 다음, 입력으로 주어지는 명령을 처리하는 프로그램을 작성하시오.

명령은 총 여섯 가지이다.

- push X: 정수 X를 큐에 넣는 연산이다.
- pop: 큐에서 가장 앞에 있는 정수를 빼고, 그 수를 출력한다. 만약 큐에 들어있는 정수가 없는 경우에는 -1을 출력한다.
- size: 큐에 들어있는 정수의 개수를 출력한다.
- empty: 큐가 비어있으면 1, 아니면 0을 출력한다.
- front: 큐의 가장 앞에 있는 정수를 출력한다. 만약 큐에 들어있는 정수가 없는 경우에는 -1을 출력한다.
- back: 큐의 가장 뒤에 있는 정수를 출력한다. 만약 큐에 들어있는 정수가 없는 경우에는 -1을 출력한다.

- https://www.acmicpc.net/problem/10845



1. 교수님 도대체 어떻게 가르쳐주신 겁니까 ..
2. 스택, 큐에 대해 복습하고 수업 들어가기 !
3. 스택 문제와 비슷한 플로우로 진행됨. 스택 문제를 풀고 이 문제를 풀었다면 난이도는 下.
4. **큐는 선입선출**

코드 전문

```
#pragma warning(disable:4996)
#pragma warning(disable:4013)
#include<stdio.h>
#include<string.h>
#include <stdlib.h>

typedef struct Node{
    int data;
    struct Node* next;
}Node;

typedef struct Queue {
    Node* front;
    Node* rear;
    int cnt;
}Quene;


void init_quene(Quene* q) {
    q->front = q->rear = NULL;
    q->cnt = 0;
}

int is_empty(Quene* q) { // empty
    return q->cnt == 0;
}

void push(Quene* q, int data) { // enquene
    Node* new = (Node*)malloc(sizeof(Node));
    new->data = data;
    new->next = NULL;

    if (is_empty(q)) {
        q->front = new;
    }
    else {
        q->rear->next = new;
    }
    q->rear = new;
    q->cnt++;
}

int pop(Quene* q) { // dequene
    int data;
    Node* a;

    if (is_empty(q)) {
        return -1;
    }
    a = q->front;
    data = a->data;
    q->front = a->next;
    free(a);
    q->cnt--;

    return data;
}

int size(Quene* q) {
    return q->cnt;
}

int front(Quene* q) {
    if (is_empty(q)) {
```

```
        return -1;
    }
    else {
        return q->front->data;
    }
}

int back(Quene* q) {
    if (is_empty(q)) {
        return -1;
    }
    else {
        return q->rear->data;
    }
}

int main() {

    Quene q;

    init_quene(&q);

    int a, num;
    char m[100] = {0};

    scanf("%d", &a);

    for (int i = 0; i < a; i++) {

        scanf("%s", m);

        if (strcmp(m, "push") == 0) {
            scanf("%d", &num);
            push(&q, num);
        }
        else if (strcmp(m, "pop") == 0) {
            printf("%d\n", pop(&q));
        }
        else if (strcmp(m, "size") == 0) {
            printf("%d\n", size(&q));
        }
        else if (strcmp(m, "empty") == 0) {
            printf("%d\n", is_empty(&q));
        }
        else if (strcmp(m, "front") == 0) {
            printf("%d\n", front(&q));
        }
        else if (strcmp(m, "back") == 0) {
            printf("%d\n", back(&q));
        }

    }

    return 0;
}
```

코드 해설

(작자 스스로를 위한 자세한 해설 🥹)

```
typedef struct Node{
    int data;
    struct Node* next;
}Node;

typedef struct Queue {
    Node* front;
    Node* rear;
    int cnt;
}Quene;
```

- 큐 구조를 실체화 시키기 위한 구조체 2개 이용.
    - Node struct : 큐 내부 작동을 위한 구조체
        - 큐 안에 들어가는 정수를 위한 data, 다음 data를 지목하기 위한 자기 참조 구조체 next
    - Queue struct : 큐 전체 작동을 위한 구조체
        - 큐 안의 front와 rear을 지목하기 위한 구조체 변수, 구조체 내의 데이터 개수를 특정하기 위한 cnt

```c
void init_quene(Quene* q) {
    q->front = q->rear = NULL;
    q->cnt = 0;
}

int is_empty(Quene* q) { // empty
    return q->cnt == 0;
}

void push(Quene* q, int data) { // enquene
    Node* new = (Node*)malloc(sizeof(Node));
    new->data = data;
    new->next = NULL;

    if (is_empty(q)) {
        q->front = new;
    }
    else {
        q->rear->next = new;
    }
    q->rear = new;
    q->cnt++;
}

int pop(Quene* q) { // dequene
    int data;
    Node* a;

    if (is_empty(q)) {
        return -1;
    }
    a = q->front;
    data = a->data;
    q->front = a->next;
    free(a);
    q->cnt--;
    return data;
}

int size(Quene* q) {
    return q->cnt;
}

int front(Quene* q) {
    if (is_empty(q)) {
        return -1;
    }
    else {
        return q->front->data;
    }
}

int back(Quene* q) {
    if (is_empty(q)) {
        return -1;
    }
    else {
        return q->rear->data;
```

```
    }
}
```

- init_queue() : 큐 초기화를 위한 함수
    - 큐의 front, rear을 모두 NULL로 만들어 주고 큐의 개수(cnt) 또한 0으로 만들어줌
- push() = enquene() : 큐에 변수(data)를 넣기 위한 함수
    - 큐에 아무것도 들어있지 않다면 q에 바로 data 입력
    - 큐에 이미 무언가 들어있다면 q의 rear에 next를 이용해서 data 연결
- pop() = dequene() : 큐에서 변수를 빼기 위한 함수
    - 큐에 무언가 들어있지 않다면 -1을 반환
    - 큐에 무언가 들어있다면 Node 변수 a를 활용하여 q의 front에 있는 data를 삭제 시켜줌(선입 선출)
- front(), back() : 큐의 맨 앞(front), 맨 뒤(rear)에 있는 변수를 출력
  -front는 구조체 변수기 때문에 정수형으로 출력 X, 그 속에 있는 data를 출력

```
int main() {
    Quene q;

    init_quene(&q);

    int a, num;
    char m[100] = {0};

    scanf("%d", &a);

    for (int i = 0; i < a; i++) {

        scanf("%s", m);

        if (strcmp(m, "push") == 0) {
            scanf("%d", &num);
            push(&q, num);
        }
        else if (strcmp(m, "pop") == 0) {
            printf("%d\n", pop(&q));
        }
        else if (strcmp(m, "size") == 0) {
            printf("%d\n", size(&q));
        }
        else if (strcmp(m, "empty") == 0) {
            printf("%d\n", is_empty(&q));
        }
        else if (strcmp(m, "front") == 0) {
            printf("%d\n", front(&q));
        }
        else if (strcmp(m, "back") == 0) {
            printf("%d\n", back(&q));
        }

    }

    return 0;
}
```

- 앞선 스택 문제와 비슷한 main 함수. 역시나 단촐하다.

📌 총평

앞선 스택 문제를 풀었다면 문제 형식을 이해하는데는 어려움이 전혀 없을 것임.

그러나 문제가 있었다면, 내가 큐를 너무 오랜만에 봤다는게 문제였음.. 그리고 자료구조 교수님이 나에게 정확한 큐를 알려주시지 않았다는 문제 .. 노션에 적어둔 필기로 도저히 풀이가 안되서 고생을 하다 결국 인터넷을 뒤졌다...

여튼 여러 이슈로 큐도 스택도 완벽하지 않으니 알고리즘 수업 들어가기 전에 제대로 보고 가기 .. !

https://code-lab1.tistory.com/6 → 참고한 블로그

Hyangim

코딩 공부 중인 대학생 🤷‍♀️

이전 포스트

백준 | 10828번 스택

0개의 댓글

댓글을 작성하세요

댓글 작성