

[학교 공부/알고리즘](#)

[알고리즘](Python) 백준 10845 - 큐

그리버 | 2023. 9. 12. 14:32 | 수정 | 삭제

유의사항			
장가계 4~6일 #100%출발확정	서유럽 4국 (영국/프랑스/스위스/이탈리아) 9~11일 #베스트셀러	서유럽 3국 (프랑스/스위스/이탈리아) 8~12일 #베스트셀러	스위스/이탈리아 #MD
419,000원~	4,280,800원~	3,049,000원~	2,690,000원~

문제
큐 개념
정답 풀이

문제

<https://www.acmicpc.net/problem/10845>

	10845번: 큐 www.acmicpc.net
---	--



주제	큐
시간 / 메모리 제한	0.5 초 / 256 MB
정답 비율	48.970%

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.

입력

- 첫째 줄에 주어지는 명령의 수 N ($1 \leq N \leq 10,000$)이 주어진다.
- 둘째 줄부터 N 개의 줄에는 명령이 하나씩 주어진다.
 - 주어지는 정수는 1보다 크거나 같고, 100,000 보다 작다.
 - 문제에 나와있지 않은 명령이 주어지는 경우는 없다.

출력

- 출력해야 하는 명령이 주어질 때마다, 한 줄에 하나씩 출력한다.

- 예제 입력 1

```
15
push 1
push 2
front
back
size
empty
pop
pop
pop
size
empty
pop
push 3
empty
front
```

- 예제 출력 1

```
1
2
2
0
1
2
-1
0
1
-1
0
3
```

큐 개념

큐(queue)는 선입선출을 뜻하는 FIFO(First In First Out) 구조이다.

먼저 들어온 데이터가 먼저 나가며, 새롭게 추가된 데이터는 큐의 맨 뒤에 추가된다.

큐의 ADT는 다음과 같다.

```
### 스택의 ADT ###
# Object
: FIFO 객체
# Operation
1. is_empty() -> Boolean
   : 큐가 비어 있으면 True, 아니면 False 반환
2. is_full() -> Boolean
   : 큐가 가득 찼으면 True, 아니면 False 반환
3. enqueue(data)
   : 큐의 맨 뒤에 데이터 삽입
4. dequeue() -> element
   : 큐의 맨 처음 데이터를 삭제하면서 반환
5. peek() -> element
   : 큐의 맨 처음 데이터를 삭제하지 않고 반환만 함
```

큐는 단순히 선형으로도 구성할 수 있지만 원형으로 구성할 수도 있다.

밑에 문제는 선형 큐로 구현하여 풀었다.

정답 풀이

시간 초과된 풀이

input()을 사용하니 시간 초과가 발생하였다.

```
class Queue:

    def __init__(self):
        self.container = list()
        self.size = 0

    def empty(self):
        if not self.container:
            return 1
        else:
```

```

        return 0

    def push(self, data):
        self.container.append(data)
        self.size += 1

    def pop(self):
        if self.size < 1:
            return -1
        self.size -= 1
        return self.container.pop(0)

    def front(self):
        if self.size < 1:
            return -1
        return self.container[0]

    def back(self):
        if self.size < 1:
            return -1
        return self.container[self.size - 1]

import sys

N = int(sys.stdin.readline())
queue = Queue()

for i in range(N):
    line = input().split()

    if line[0] == 'push':
        queue.push(line[1])

    elif line[0] == 'pop':
        print(queue.pop())

    elif line[0] == 'size':
        print(queue.size)

    elif line[0] == 'empty':
        print(queue.empty())

    elif line[0] == 'front':
        print(queue.front())

    elif line[0] == 'back':
        print(queue.back())

```

정답 풀이

input()을 sys.stdin.readline()로 대체하여 작성했더니 통과했다.

```

class Queue:

    def __init__(self):
        self.container = list()
        self.size = 0

    def empty(self):
        if not self.container:
            return 1
        else:
            return 0

    def push(self, data):
        self.container.append(data)
        self.size += 1

    def pop(self):
        if self.size < 1:
            return -1
        self.size -= 1
        return self.container.pop(0)

    def front(self):
        if self.size < 1:
            return -1
        return self.container[0]

    def back(self):
        if self.size < 1:
            return -1
        return self.container[self.size - 1]

import sys

N = int(sys.stdin.readline())
queue = Queue()

for i in range(N):
    line = sys.stdin.readline().split()

```

```
if line[0] == 'push':
    queue.push(line[1])

elif line[0] == 'pop':
    print(queue.pop())

elif line[0] == 'size':
    print(queue.size)

elif line[0] == 'empty':
    print(queue.empty())

elif line[0] == 'front':
    print(queue.front())

elif line[0] == 'back':
    print(queue.back())
```

♡ 공감

🔗

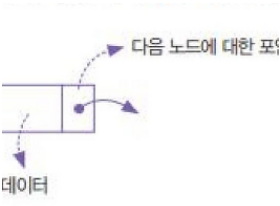
📊

☰

'학교 공부 > 알고리즘' 카테고리의 다른 글

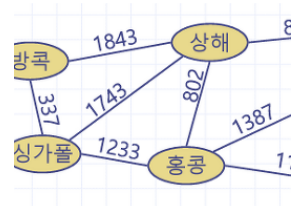
[알고리즘](Python) 리스트 자료구조 (0)	2023.09.11
[알고리즘](C언어) 그래프 알고리즘 (2)	2022.11.21
[알고리즘](C언어) 분할통치법 - 합병정렬 (0)	2022.11.10
[알고리즘](C언어) 해싱을 이용한 사전의 해시테이블 구현 (2)	2022.11.07
[알고리즘](C언어) 트리 탐색을 이용한 사전의 트리 구현 - AVL 트리 (0)	2022.11.07

'학교 공부/알고리즘' Related Articles

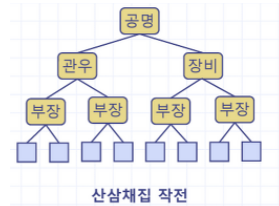


다음 노드에 대한 포인터

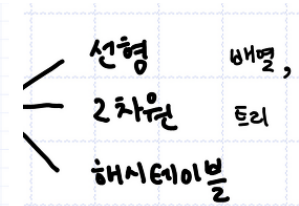
데이터



방콕 1843 상해 802
337 1743 1233 1387
싱가포르 1743 홍콩 1233



공명
관우 장비
부장 부장 부장 부장
산삼채집 작전



선형 배열,
2차원 트리
해시테이블

[알고리즘](Python) 리스트 자료구조

[알고리즘](C언어) 그래프 알고리즘

[알고리즘](C언어) 분할통치법 - 합병정렬

[알고리즘](C언어) 해싱을 이용한 사전의 해시테이블 구현

☐ Secret

안녕하세요! 어떤 댓글이든 환영합니다! 🍷

댓글달기