

DEQUE(이)란 무엇인가?

WED, MAR 27, 2024

윤태영

CONTENTS

- deque의 발음
- deque의 기본 개념
- deque, list와 무엇이 다른가?
- deque의 사용법
- 실제 문제에서의 적용 사례

DEQUE의 발음

'DECK' & 'DQ'

무엇이 올바른 발음인가?

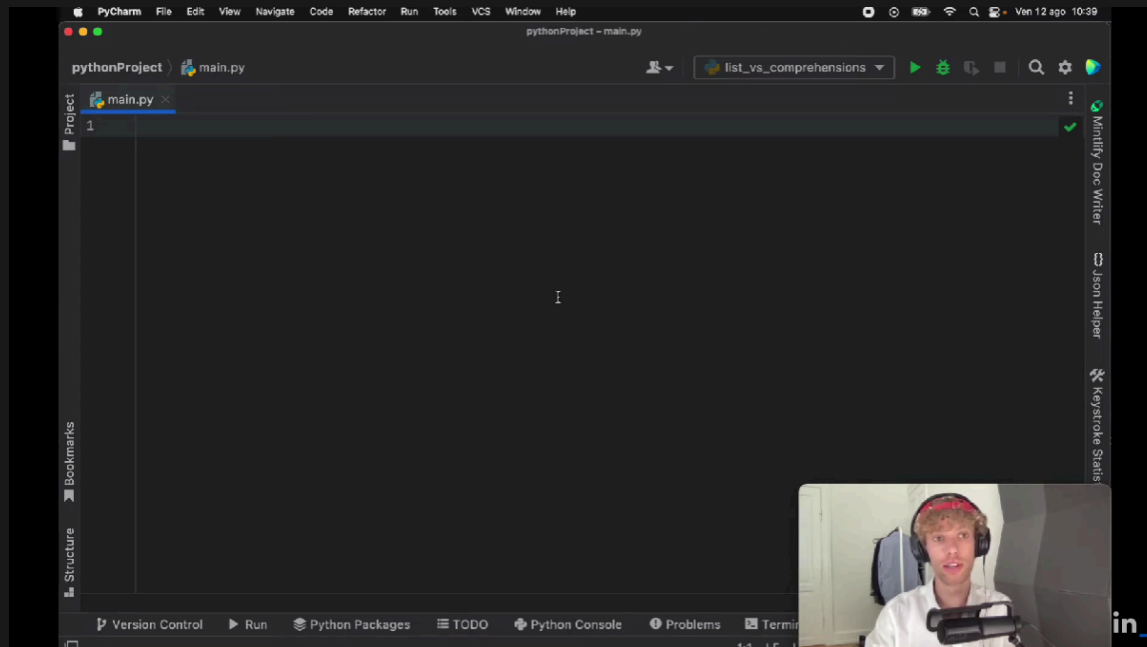
네이티브들은 어떻게 발음할까?

DEQUE의 발음



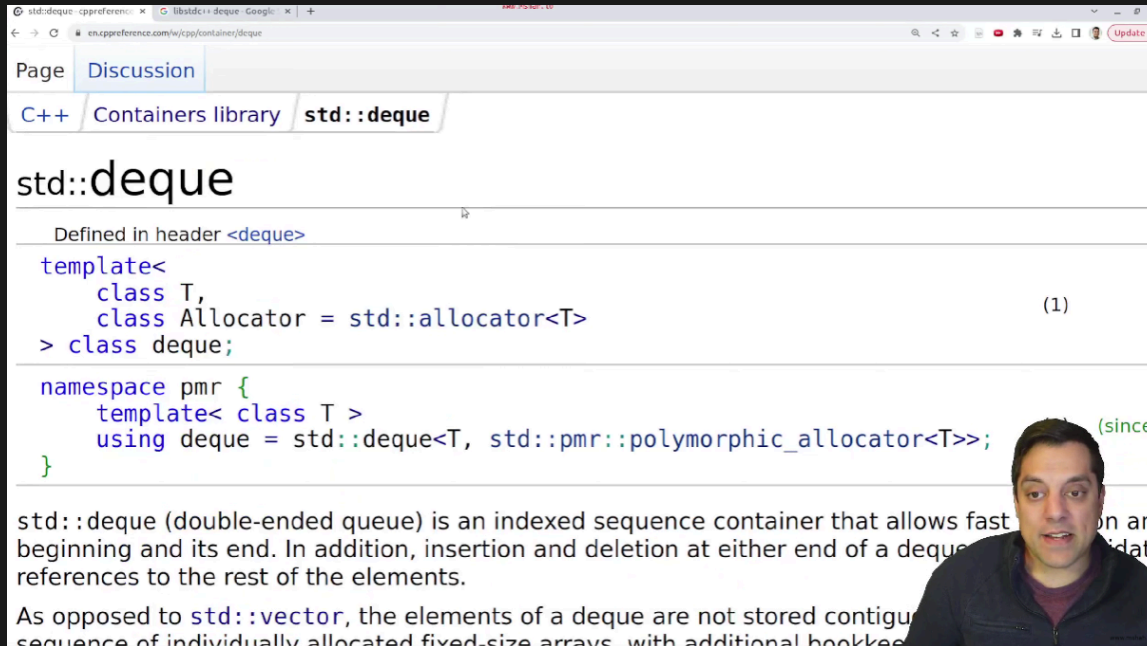
"We're going to be talking about **DECK**."

DEQUE의 발음



"In today's lesson we're going to be covering DECK."

DEQUE의 발음



The screenshot shows the C++ reference website for `std::deque`. The page title is `std::deque`, and it is defined in the header `<deque>`. The code snippet shows the template definition for `std::deque` and a `pmr` namespace specialization. A video overlay of a man speaking is visible in the bottom right corner of the screenshot.

```
template<
    class T,
    class Allocator = std::allocator<T>
> class deque;

namespace pmr {
    template< class T >
        using deque = std::deque<T, std::pmr::polymorphic_allocator<T>>;
}
```

`std::deque` (double-ended queue) is an indexed sequence container that allows fast insertion and deletion at either end of a deque. It also allows constant time access to the beginning and its end. In addition, insertion and deletion at either end of a deque does not require moving references to the rest of the elements.

As opposed to `std::vector`, the elements of a deque are not stored contiguous. Instead, they are stored in a sequence of individually allocated fixed-size arrays, with additional bookkeeping to maintain the sequence.

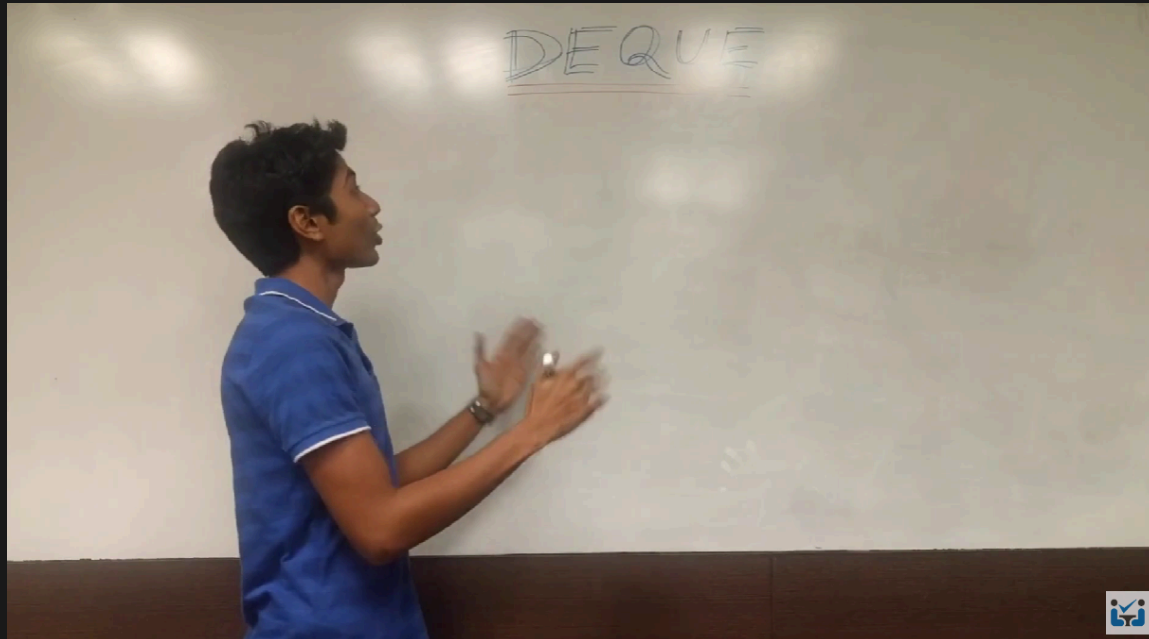
"So what we see here with the DECK data structure is..."

그러면

DECK

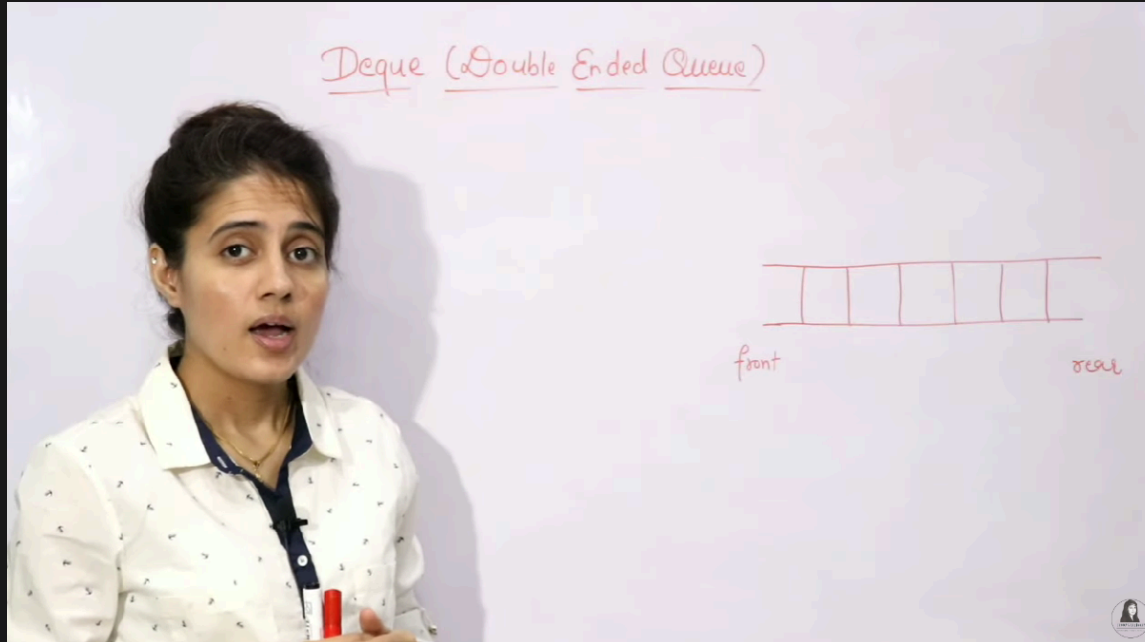
이 올바른 발음인 것인가?

DEQUE의 발음



"Hi everyone. We will talk about DQ."

DEQUE의 발음



"In this lecture we are going to talk about DQ."

DEQUE의 발음

'DECK' & 'DQ'

둘 다 맞다!

DEQUEUE의 기본 개념

Double-Ended-QUEUE

=> 양 끝에서 삽입과 삭제가 가능하다는 특징을 갖고 있다.

근데 그거 LIST에서 다 되는 건데요?

```
1 Li = [1, 2, 3, 4, 5]
2
3 # 앞에서 삭제 & 삽입
4 Li.pop(0)           # [2, 3, 4, 5]
5 Li.insert(0, 1)     # [1, 2, 3, 4, 5]
6
7 # 뒤에서 삭제 & 삽입
8 Li.pop()            # [1, 2, 3, 4]
9 Li.append(5)        # [1, 2, 3, 4, 5]
```

DEQUE, LIST와 무엇이 다른가?

LIST에서의 삽입 & 삭제

LIST에서의 삽입 & 삭제



LIST에서의 삽입 & 삭제



앞에서 삽입: $O(n)$

LIST에서의 삽입 & 삭제



앞에서 삭제: $O(n)$

LIST에서의 삽입 & 삭제



뒤에서 삽입: $O(1)$

LIST에서의 삽입 & 삭제



뒤에서 삭제: $O(1)$

DEQUE에서의 삽입 & 삭제



DEQUE에서의 삽입 & 삭제



앞에서 삽입: $O(1)$

DEQUE에서의 삽입 & 삭제



앞에서 삭제: $O(1)$

DEQUE에서의 삽입 & 삭제



뒤에서 삽입: $O(1)$

DEQUE에서의 삽입 & 삭제



뒤에서 삭제: $O(1)$

Deque,

양방향 연결 리스트

로 구현되어져 있다...!

DEQUE의 사용법

```
1 from collections import deque
2
3 li = [1, 2, 3, 4, 5]
4 dq = deque(li)
```

DEQUE의 사용법

```
1 from collections import deque
2
3 # li = [1, 2, 3, 4, 5]
4 # dq = deque(li)
5 dq = deque([1, 2, 3, 4, 5])
```

DEQUE의 사용법

```
1 from collections import deque
2
3 # li = [1, 2, 3, 4, 5]
4 # dq = deque(li)
5 dq = deque([1, 2, 3, 4, 5])
6
7 # 앞에서 삭제 & 삽입
8 dq.popleft()          # deque([2, 3, 4, 5])
9 dq.appendleft(1)      # deque([1, 2, 3, 4, 5])
10
11 # 뒤에서 삭제 & 삽입
12 dq.pop()             # deque([1, 2, 3, 4])
13 dq.append(5)         # deque([1, 2, 3, 4, 5])
```

DEQUE의 사용법

```
1 from collections import deque
2
3 # li = [1, 2, 3, 4, 5]
4 # dq = deque(li)
5 dq = deque([1, 2, 3, 4, 5])
6
7 # 앞에서 삭제 & 삽입
8 dq.popleft()          # deque([2, 3, 4, 5])
9 dq.appendleft(1)      # deque([1, 2, 3, 4, 5])
10
11 # 뒤에서 삭제 & 삽입
12 dq.pop()             # deque([1, 2, 3, 4])
13 dq.append(5)         # deque([1, 2, 3, 4, 5])
14
15 # 돌려라!
```

실제 문제에서의 적용 사례

4 1158번

제출

맞힌 사람

숏코딩

재채점 결과

채점 현황

내 제출

난이도 기여

강의

질문 게시판 (20)

타이머

요세푸스 문제

성공

☆

시간 제한

메모리 제한

제출

정답

맞힌 사람

정답 비율

2 초

256 MB

109540

55070

38756

49.055%

문제

요세푸스 문제는 다음과 같다.

1번부터 N번까지 N명의 사람이 원을 이루면서 앉아있고, 양의 정수 $K(≤ N)$ 가 주어진다. 이제 순서대로 K번째 사람을 제거한다. 한 사람이 제거되면 남은 사람들로 이루어진 원을 따라 이 과정을 계속해 나간다. 이 과정은 N명의 사람이 모두 제거될 때까지 계속된다. 원에서 사람들이 제거되는 순서를 (N, K)-요세푸스 순열이라고 한다. 예를 들어 (7, 3)-요세푸스 순열은 <3, 6, 2, 7, 5, 1, 4>이다.

N과 K가 주어지면 (N, K)-요세푸스 순열을 구하는 프로그램을 작성하시오.

입력

첫째 줄에 N과 K가 빈 칸을 사이에 두고 순서대로 주어진다. ($1 ≤ K ≤ N ≤ 5,000$)

출력

예제와 같이 요세푸스 순열을 출력한다.

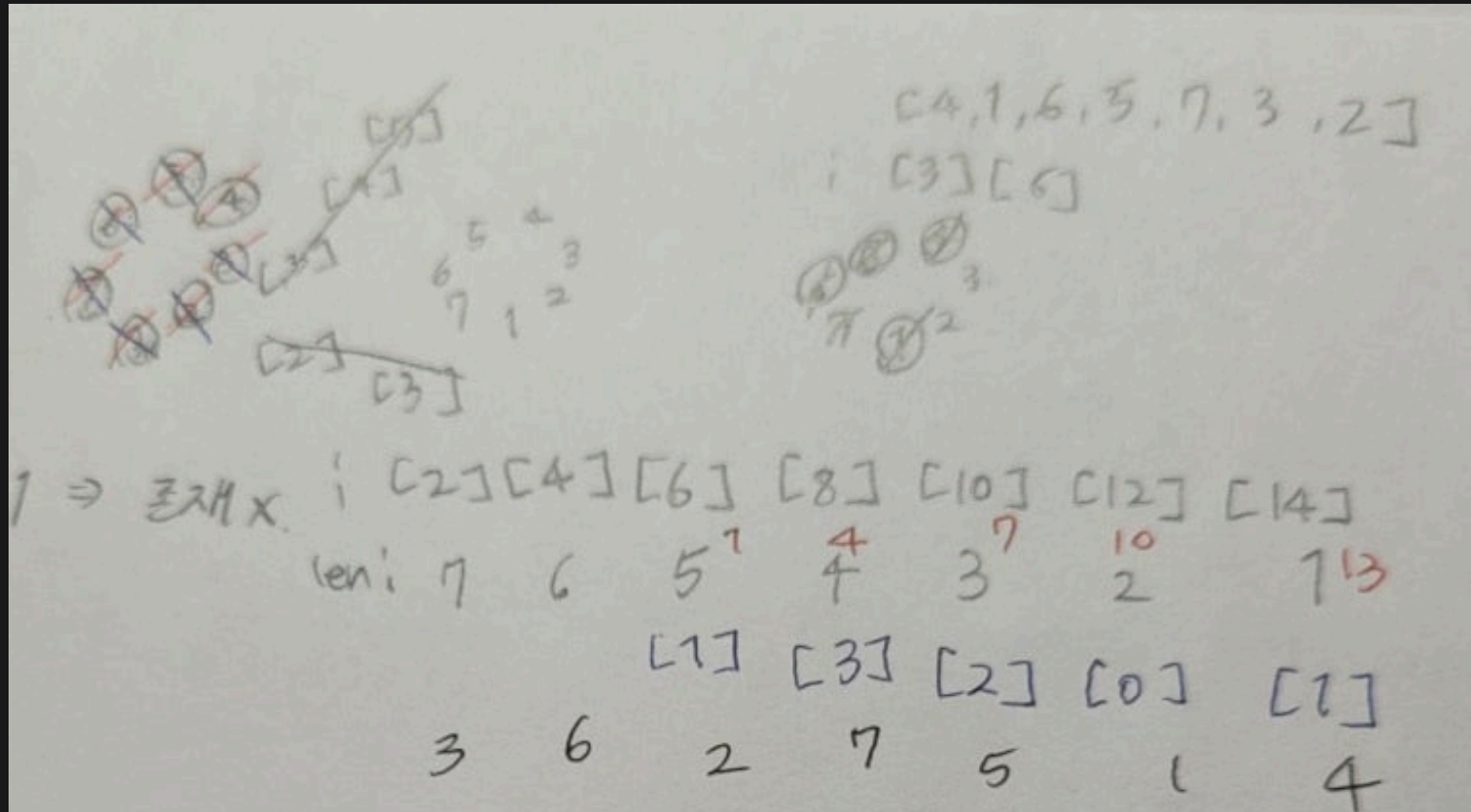
예제 입력 1 복사

7 3

예제 출력 1 복사

<3, 6, 2, 7, 5, 1, 4>

실제 문제에서의 적용 사례



2022년 5월 초, 실버3 유저의 고뇌의 흔적...!

실제 문제에서의 적용 사례

```
1 import sys
2 input = sys.stdin.readline
3 N, K = map(int, input().split())
4 nums = list(range(1, N+1))
5 permutation = []
6 i = K-1
7
8 for _ in range(N) :
9     while True :
10         if i >= len(nums) :
11             i -= len(nums)
12         else :
13             break
14     permutation.append(nums[i])
15     del nums[i]
16     i += K-1
17
18 print("<", end = "")
19 for i in range(len(permutation)) :
20     if i != len(permutation)-1 :
21         print(permutation[i], end = ", ")
22     else :
23         print(permutation[i], end = "")
24 print(">", end = "")
```



2022년 5월산 코드, 정말 읽기 싫게 생겼다.

실제 문제에서의 적용 사례

```
1 import sys; input = sys.stdin.readline
2 from collections import deque
3
4 def solution(N: int, K: int) -> str:
5     ans = []
6
7     people = deque([str(i+1) for i in range(N)])
8     while people:
9         people.rotate(-(K-1))
10        ans.append(people.popleft())
11
12    return '<' + ', '.join(ans) + '>'
13
14
15 if __name__ == "__main__":
16     N, K = map(int, input().split())
17     print(solution(N, K))
```

2024년 3월산 코드, 깔끔해졌다!

실제 문제에서의 적용 사례

제출 번호	아이디	문제	결과	메모리	시간	언어	코드 길이	제출한 시간
75773309	yoomen	 요세푸스 문제	맞았습니다!!	34148 KB	68 ms	Python 3 / 수정	267 B	39초 전
43447209	yoomen	 요세푸스 문제	맞았습니다!!	30840 KB	84 ms	Python 3 / 수정	520 B	1년 전

2022년 5월산 코드 대비 20% 빨라진 것을 확인할 수 있었다.

감사합니다