

Queue ($\exists \pi$)

큐

- 특성

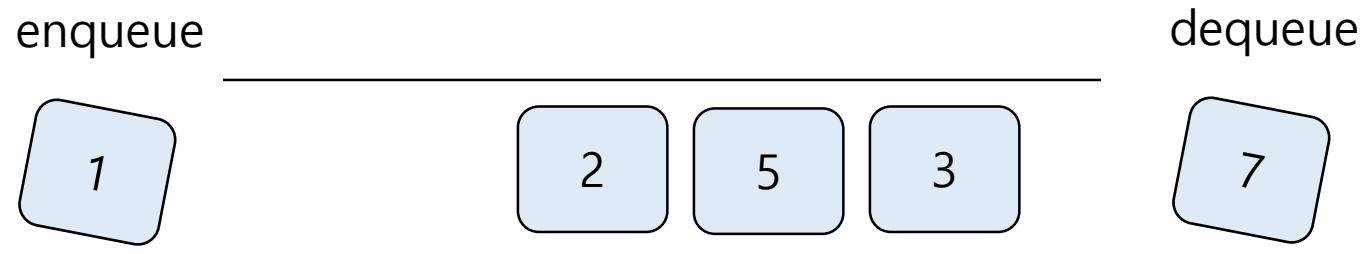
- Similar with a pipe
- First In First Out

- 주요 연산

- `insert()`
- `remove()`

- 활용용

- 스케줄링 (FCFS, round-robin, ...)
- ...



큐 구현

- *doubly linked list* 또는 *circular linked list* 로 구현
 - *enqueue()* : *insert_back()*
 - *dequeue()* : *delete_front()*
- 파이썬 리스트로 구현
 - *enqueue()*
 - *dequeue()*

큐 구현 - 양방향 연결 리스트 활용

- 클래스 Queue 정의

- one member variables

- queue : 아이템들을 저장할 리스트. 객체 생성 시에 빈 리스트로 초기화

- 생성자, enqueue(), dequeue(), print_queue()

```
import Dlist
class Queue:
    def __init__(self):
        self.queue = Dlist.Dlist()
```

큐 구현 - 양방향 연결 리스트 활용

- 클래스 Queue 정의

- enqueue() & dequeue()

```
class Queue :
```

```
...
```

```
def enqueue(self, item):
```

```
    # 리스트의 가장 뒤에 추가....
```

```
def dequeue(self):
```

```
    # 리스트의 가장 앞에서 제거....
```

큐 구현 - 양방향 연결 리스트 활용

- 클래스 Queue 정의

- enqueue() & dequeue()

```
class Queue :
```

```
...
```

```
def enqueue(self, item):
```

```
    self.queue.insert_back(item)
```

```
def dequeue(self):
```

```
    return (self.queue.delete_front()) # delete_front() 함수는 삭제한 아이템을 반환해야 함.. 삭제 실패 시에는 None 반환
```

큐 구현 - 양방향 연결 리스트 활용

- 클래스 Queue 정의

- `print_queue()`

```
class Queue :
```

```
...
```

```
def print_queue(self):
```

```
    self.queue.print_list()
```

큐 활용 프로그램 /

```
q = Queue()
```

```
q.enqueue('mango')
```

```
q.enqueue('apple')
```

```
q.enqueue('orange')
```

```
q.print_queue()
```

```
q.dequeue()
```

```
q.print_queue()
```

```
q.dequeue()
```

```
q.print_queue()
```

```
q.dequeue()
```

```
q.print_queue()
```

```
q.dequeue()
```

```
q.print_queue()
```


활용 분야

- CPU의 태스크 스케줄링(Task Scheduling)
- 네트워크 프린터 큐
- 콜 센터의 전화 서비스 처리
- 그래프의 너비 우선 탐색 (Breadth-First Search)
- ...

Homework candidate

- 큐에서 특정 아이템을 검색하여 발견되면 *found*, 없으면 *not found*를 출력하는 *search()* 함수 추가 구현하기

Appendix

- Circular Q (원형 큐)
 - front & rear
 - 크기가 고정됨
 - message mailbox 등
- DEQueue (Double Ended Queue), 덱
 - 전면과 후면에서 모두 삽입, 삭제 가능
- 우선순위 큐
 - item들이 각기 다른 우선순위를 가질 수 있음.
 - dequeue시 가장 우선순위가 높은 item을 꺼냄.

큐 구현 - 파이썬 리스트 활용

- 클래스 Queue 정의

- one member variables

- *queue* : 아이템들을 저장할 리스트. 객체 생성 시에 빈 리스트로 초기화

- 생성자, *enqueue()*, *dequeue()*, *print_queue()*

```
class Queue:  
    def __init__(self):  
        self.queue = []
```

큐 구현 - 파이썬 리스트 활용

- 클래스 Queue 정의

- enqueue() & dequeue()

```
class Queue :
```

```
...
```

```
def enqueue(self, item):
```

```
    # 리스트의 가장 앞에 추가....
```

```
def dequeue(self):
```

```
    # 리스트의 가장 뒤에서 제거....
```

파이썬 - 리스트 관련 메서드

- `insert(index, item)`

- 리스트의 `index` 위치에 `item` 삽입
- `append()`는 끝에만 삽입하는데 `insert()`는 임의의 위치에 삽입
- `mylist = [1, 2, 3]`
- `mylist.insert(0, 4)`
- `print(mylist)`
 - `[4, 1, 2, 3]` 출력됨

파이썬 — 리스트 관련 메서드

- `pop(index)`
 - `index` 위치의 `item`을 제거 & `return`
 - `index`가 없으면 리스트의 끝에서..
 - `index`가 `-1`인 경우도 리스트의 끝에서..
 - `mylist = [1, 2, 3, 4]`
 - `print(mylist.pop(0))`
 - 1 출력됨
 - `print(mylist.pop())`
 - 4 출력됨
 - `print(mylist.pop(-1))`
 - 3 출력됨

큐 구현 - 파이썬 리스트 활용

- 클래스 Queue 정의

- enqueue() & dequeue()

```
class Queue :
```

```
...
```

```
def enqueue(self, item):
```

```
    self.queue.insert(0, item)
```

```
def dequeue(self):
```

```
    if (len(self.queue) == 0):
```

```
        return None;
```

```
    return (self.queue.pop())
```


큐 구현 - 파이썬 리스트 활용

- 클래스 Queue 정의

- `print_queue()`

```
class Queue :
```

```
...
```

```
def print_queue(self):
```

```
    print(self.queue)
```