

Queue Reconstruction by Height

# Index	406
📅 CreatedAt	@September 28, 2022
👤 Person	A Ally Hyeseong Kim
☰ Status	In Progress
☰ Tags	Greedy Python
📅 UpdatedAt	@September 28, 2022

References

LeetCode - The World's Leading Online Programming Learning Platform
Level up your coding skills and quickly land a job. This is the best place to expand your knowledge and get prepared for your next interview.

<https://leetcode.com/problems/queue-reconstruction-by-height/>



파이썬 알고리즘 인터뷰

2021 세종도서 학술부문 선정작. 현업과 실무에 유용한 주요 알고리즘 이론을 깊숙이 이해하고, 파이썬의 핵심 기능과 문법까지 상세하게 이해할 수 있는 취업용 코딩 테스트를 위한 완벽 가이드다. 200여 개가 넘는...

<https://www.aladin.co.kr/shop/wproduct.aspx?ItemId=245495826>



References

- [1. Greedy Algorithm](#)
- [2. HeapQueue](#)

1. Greedy Algorithm

1.1. Solution 1

```
class Solution:
    def reconstructQueue(self, people: List[List[int]]) -> List[List[int]]:

        people = sorted(people, key=lambda p: [p[1], p[0]])
        answer = []

        zeros = 0
        for i, p in enumerate(people):
            if p[1] == 0:
                zeros += 1
                answer.append(p)
            else:
                count = 0
                for j in range(len(answer)):
                    if answer[j][0] >= p[0]:
                        count += 1
                if count > p[1]:
                    answer = answer[:j] + [p] + answer[j:]
                    count = 0
                    break
            if count:
```

```

        answer.append(p)

    return answer

```

1.2. Solution 2

```

class Solution:
    def reconstructQueue(self, people: List[List[int]]) -> List[List[int]]:

        people = sorted(people, key=lambda p: [-p[0], p[0]])
        answer = []

        for p in people:
            answer.insert(p[1], [p[0], p[1]])

        return answer

```

- `list.insert(index, element)`: 해당 index에 element 넣기

2. HeapQueue

```

class Solution:
    def reconstructQueue(self, people: List[List[int]]) -> List[List[int]]:

        heap = []
        for p in people:
            heapq.heappush(heap, (-p[0], p[1]))

        answer = []
        while heap:
            p = heapq.heappop(heap)
            answer.insert(p[1], [-p[0], p[1]])

        return answer

```

- `heapq`: 우선순위 큐 → Python은 `Min Heap` 만 지원
 - Python에서 sort: Tim Sort, $O(n \log n)$
 - Python에서 heap sort: $O(\log n)$
- `heapq.heappush(heap, element)`
- `heapq.heappop(heap)`