

예시)  $N = 3$

arr =

13	0	50	0번 일
12	70	90	1번 일
25	60	100	2번 일
0번 사람	1번 사람	2번 사람	

최대 값 구하기

최대 값을 가지는 순열 찾기

0번 사람, 1번 사람, 2번 사람으로 순열 생성!

[0, 1, 2]  
0번 일 1번 일 2번 일  
[0, 2, 1]  
[1, 0, 2]  
⋮

[SWEA]5205 킁정렬

예시)

2	2	4	1	3
---	---	---	---	---

i j Pivot(정렬대상)

- i) Pivot보다 작은 수 → 이 중 마지막에 위치한 수의 index = j
- ii) Pivot 보다 크거나 같은 수
- iii) 아직 check 하지 않은 수 → 이 중 첫번째에 위치한 수의 index = j

2	2	4	1	3
---	---	---	---	---

i j

2	2	4	1	3
---	---	---	---	---

i j

2	2	4	1	3
---	---	---	---	---

i j

2	2	1	4	3
---	---	---	---	---

i j

2	2	1	3	4
---	---	---	---	---

i+1

- j는 항상 1씩 증가한다.
  - i는 arr[j]가 Pivot보다 작을 때 증가 (i 값 ←→ j 값)
  - j가 Pivot 전까지 도달하면 반복 종료 & i+1과 Pivot 변경
- Pivot보다 큰 값 중 가장 앞에 위치한 수

[SWEA]5207 이진탐색

예시)  $N = 10$   
 $M = 2$

list\_a =

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

mid

S e

list\_b =

6	5
---	---

- 이진탐색 (mid 값과 비교)
- mid 값이 더 크면 왼쪽 탐색
  - mid 값이 더 작으면 오른쪽 탐색

- 조건) 탐색 성공!
- 조건) 좌·우 번갈아서 탐색 → 어느 쪽 탐색하는지 기록



[SWEA]5208 전기버스2

$0 \leq |s| \leq N-1$   $N=10$

bus가 여기에 도착하면 종료

arr = 

	2	1	3	2	2	5	4	2	1
--	---	---	---	---	---	---	---	---	---

      N-1 번째 인덱스 (종료조건)

**배터리를 교체**  
**배터리를 교체하지 않는 경우**

arr[i]만큼 이동가능  $\rightarrow$  이동하면  $power -= 1$   
 $power$  power가 0이되면 종료 (종료조건)

**[SWEA] 5309 최소생산비용**

예시)  $N = 3$  arr =

73	21	21
11	59	40
24	31	83

A B C

1 → 0번제품  
2 → 1번제품  
3 → 2번제품

0번, 1번, 2번 제품으로 순열 생성

[0번, 1번, 2번]	
A B C	→ 순열[0]은 A공장 제품
[0번, 2번, 1번]	순열[1]은 B공장 제품
⋮	순열[2]은 C공장 제품

## 최대값을 가지는 순열 찾기