Greedy Algorithm.

1. 완전 탐색과 그리디

	완탐 / DP	Greedy
공통점.	(분할 정복) 여러 조각으로 쪼개고, 각 단계마다 답의 한 부분을 만들어 감.	
차이점	모든 선택지를 고려해보고. 그 중 전체 답이 가장 좋은 방법만을 선택	각 단계마다 지금 당장 좋은 방법만을 선택 (지금 선택이 앞으로 남은 선택들에 어떤 영향을 미칠지 고려하지 X)

2. 그리디 알고리즘이 사용되는 경우.

① 탐욕법을 사용해도 항상 최적해를 구할 수 있는 문제일 경우,

동적 계획법보다 수행시간이 훨씬 빠르기 때문에 유용하다.

② 시간이나 공간적 제약으로 인해 다른 방법으로 최적해를 찾기 어려운 경우.

최적해 대신 적당히 괜찮은 값 (근사해)을 찾는 것으로 타협.

(탐욕법으로 최적해를 찾을 수 있는 많은 문제들은 동적 계획법으로도 풀 수 있다. 탐욕법으로 최적해를 찾을 수 있다 = 지금 한 단계만을 고려해도 답을 찾을 수 있다.

따라서 모든 단계를 고려하는 동적 계획법이 답이 틀릴 수가 없다. 그럼에도 탐욕법을 사용하는 이유는. 동적 계획법에 필요한 메모리나 시간이 과도하게 크기 때문)

3. 알고리즘의 정당성 증명 : "탐욕적 선택 속성이 성립한다"

(동적 계획법처럼) 답의 모든 부분을 고려하지 않고 탐욕적으로만 선택하더라도 최적해를 구할 수 있다.

⇒ ∴ 탐욕적인 선택을 해서 '손해'를 볼 일이 없다.

예) 회의실 예약.

1) 목록 S에 남은 회의 중 가장 일찍 끝나는 회의 S_{min}을 선택한다.

2) S_{min}과 겹치는 회의를 S에서 모두 지운다.

3) S가 텅 빌 때까지 반복한다.

'탐욕적 선택 속성이 성립한다' ⇔ 다음 조건이 성립한다.

'가장 종료 시간이 빠른 회의(S_{min})를 포함하는 최적해가 반드시 존재한다.'

증명)

S'

S의 최적해 中 S_{min}을 포함하지 않는 답이 존재

이 답은 서로 겹치지 않는 회의의 목록 임.

이 목록에서 첫번째로 개최되는 회의를 지우고 S_{min}을 대신 추가해서 새로운 목록을 만든다.

S_{min}은 S에서 가장 일찍 끝나는 회의이기 때문에 지워진 회의는 S_{min} 보다 일찍 끝날 수 없다.

따라서 S'의 가장 첫번째 회의를 S_{min}으로 대체하여 새로 만든 목록 역시 최적해 중 하나가 된다.

서로 겹치지 않음.

$$S' = \{ \underbrace{a, b, c}, \dots \}$$

S_{min}의 끝나는 시간 < a의 끝나는 시간.

$S'' = \{ S_{min}, b, C, \dots \}$ → 마찬 가지로 최적해.