

ICPC Sinchon



2022 Winter Algorithm Camp

4회차. 그리디 알고리즘

서강대학교 김성현

2022 Winter Algorithm Camp

4회차. 그리디 알고리즘

목차

1. 그리디의 개념
2. 기초적인 예시와 그리디의
한계
3. 처리 순서 고려 - 예시
4. 제약에 대한 고려 - 예시
5. 최적 전략의 반복 - 예시
6. 그리디의 발상

2022 Winter Algorithm Camp

Greedy Algorithm - 개론

* 그리디 알고리즘은 무엇인가?

- 부분적인 최적 전략을 반복적으로 취하는 알고리즘
- 당장 보이는 이득을 좇아가는 것도 이 알고리즘의 일부

Greedy Algorithm - 개론

* 그리디 알고리즘은 무엇인가?

- 부분적인 최적 전략을 반복적으로 취하는 알고리즘
- 당장 보이는 이득을 좇아가는 것도 이 알고리즘의 일부

* 어떤 문제가 그리디인가?

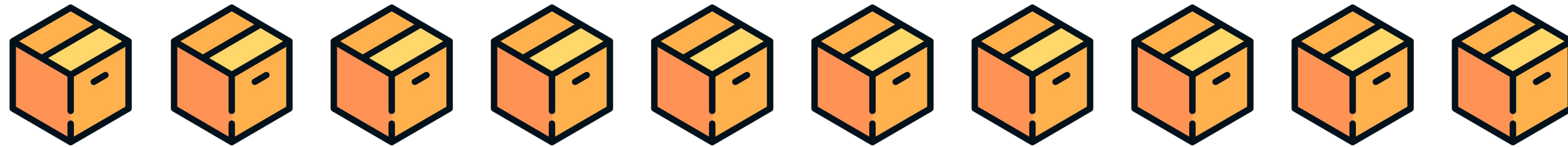
- 부분적인 최적 전략을 반복적으로 취해서 답을 구할 수 있는 문제
- 즉 부분 문제에 대해서 구한 최적해들이 모두 전체 문제의 최적해의 일부인 문제

Greedy Algorithm - 개론

* 무엇을 고려해야 하는가?

- 전체 문제를 얼마큼씩 처리할 것인가?
- 부분 문제의 최적 전략은 무엇인가?
- 그 전략 실행의 제약사항은 없는가? 있다면 어떻게 충족시킬 것인가?
- 부분 문제들에 대해 특정 순서로 최적 전략들을 반복하면 정말 전체 문제의 최적해가 나오는가?

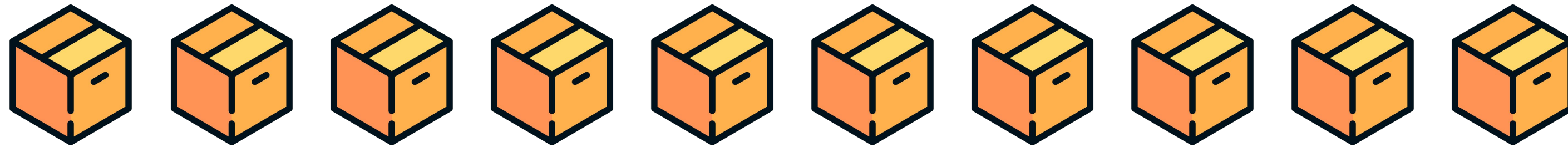
Greedy Algorithm - 예시



상자 번호	1	2	3	4	5	6	7	8	9	10
상자 가치	3	12	7	5	1	23	19	4	3	15

* 상자들 중 최대 3개를 골라 그 가치 합이 가장 크게 하려면?

Greedy Algorithm - 예시

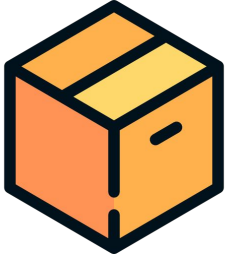
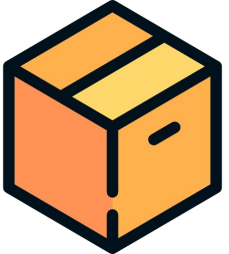
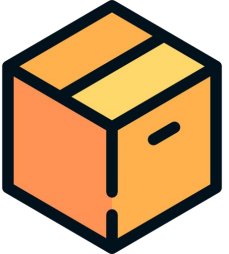
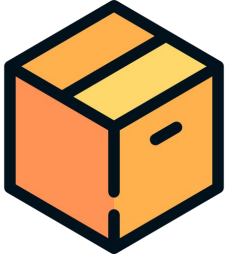
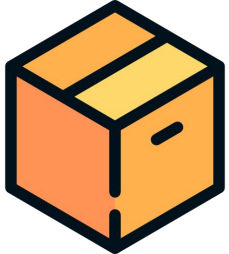
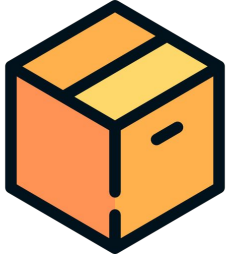
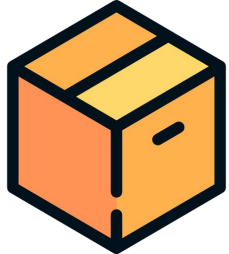
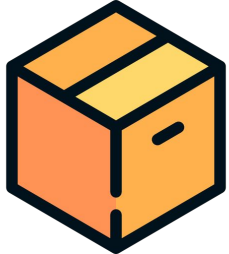
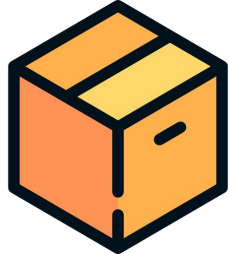
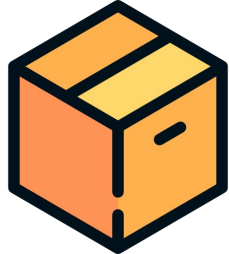


상자 번호	1	2	3	4	5	6	7	8	9	10
상자 가치	3	12	7	5	1	23	19	4	3	15

* 상자들 중 최대 3개를 골라 그 가치 합이 가장 크게 하려면?

- 가장 가치가 큰 3개를 고르면 된다
- 아직 택하지 않은 상자들 중 가치가 가장 큰 상자를 고르는 전략을 3번 반복
- 한번에 한 개씩 고르면서, 한 개 골랐을 때 가장 이득이 되는 것을 찾는 것

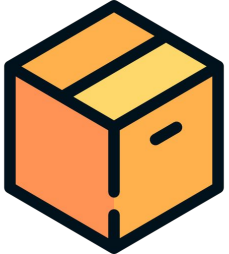
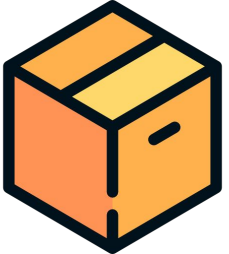
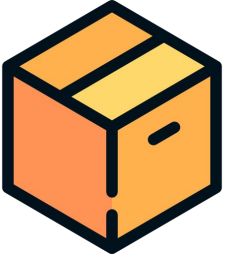
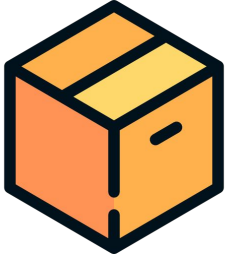
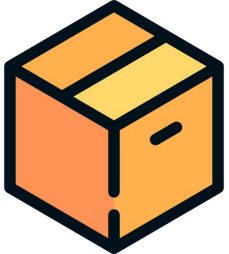
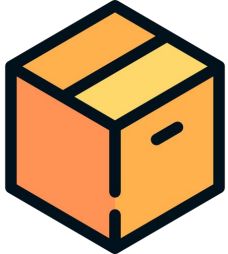
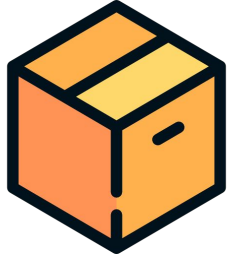
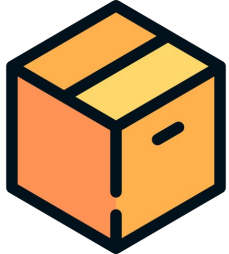
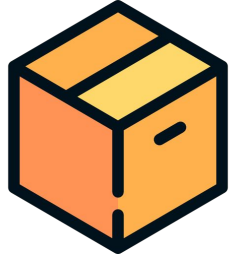
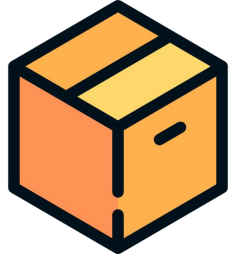
Greedy Algorithm - 예시

										
상자 번호	1	2	3	4	5	6	7	8	9	10
상자 가치	3	12	7	5	1	23	19	4	3	15

* 상자들 중 최대 3개를 골라 그 가치 합이 가장 크게 하려면?

- 가장 가치가 큰 3개를 고르면 된다
- 아직 택하지 않은 상자들 중 가치가 가장 큰 상자를 고르는 전략을 3번 반복
- 한번에 한 개씩 고르면서, 한 개 골랐을 때 가장 이득이 되는 것을 찾는 것

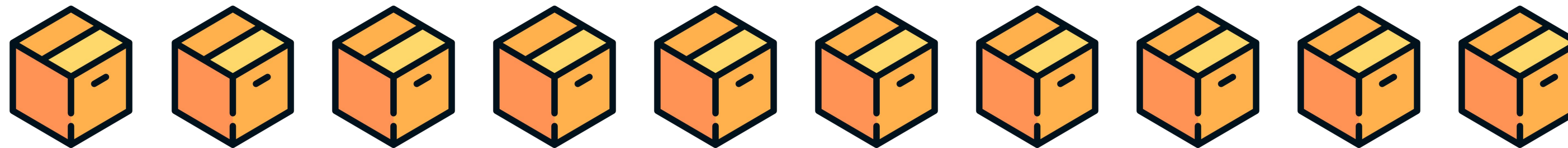
Greedy Algorithm - 예시

										
상자 번호	1	2	3	4	5	6	7	8	9	10
상자 가치	-9	-3	17	-5	-11	23	-8	-4	-3	-6

* 상자들 중 최대 3개를 골라 그 가치 합이 가장 크게 하려면?

- 이번에는 상자를 3개 고르는 게 최적이지 아님
- 남은 상자들 중 가치가 양수이며 가장 가치가 큰 상자를 고르는 전략을 3번 반복
- 만약 남은 상자들 중 가치가 양수인 상자가 없으면 고르지 않는다

Greedy Algorithm - 예시



상자 번호

1

2

3

4

5

6

7

8

9

10

상자 가치

-9

-3

17

-5

-11

23

-8

-4

-3

-6

* 부분 문제는?

- 남은 상자 중
상자를 1개씩 고름

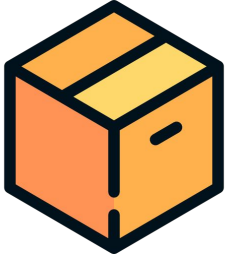
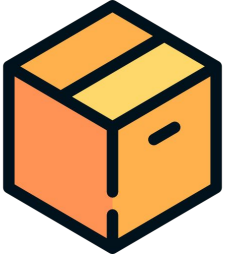
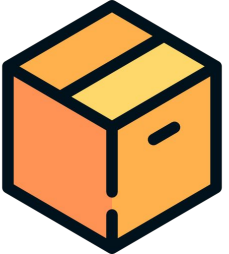
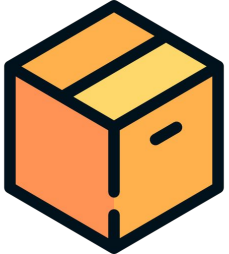
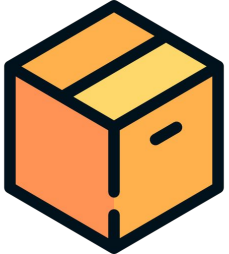
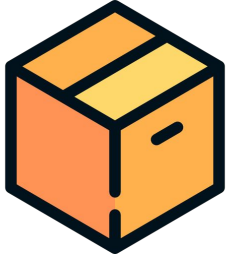
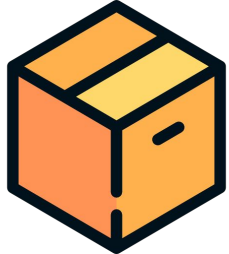
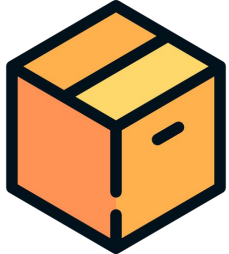
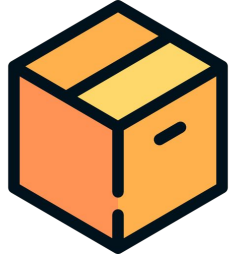
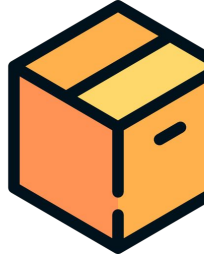
* 최적 전략은?

- 가치가 가장 큰
상자를 고름

* 제약 사항은?

- 상자의 가치가 양수일
경우에만 고름

Greedy Algorithm - 예시

										
상자 번호	1	2	3	4	5	6	7	8	9	10
상자 가치	-9	-3	17	-5	-11	23	-8	-4	-3	-6

* 부분 문제는?

- 남은 상자 중
상자를 1개씩 고름

* 최적 전략은?

- 가치가 가장 큰
상자를 고름

* 제약 사항은?

- 상자의 가치가 양수일
경우에만 고름

* 진짜 최적인가?

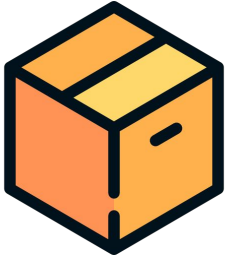
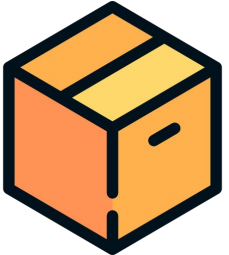
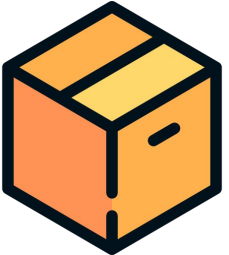
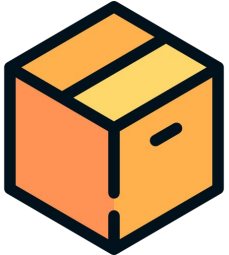
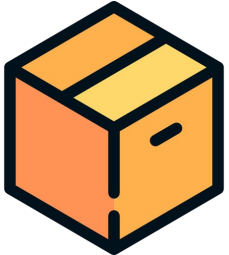
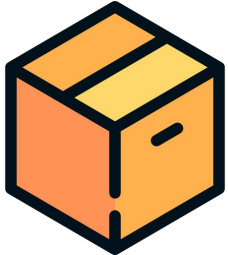
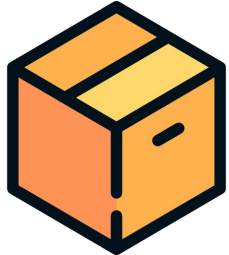
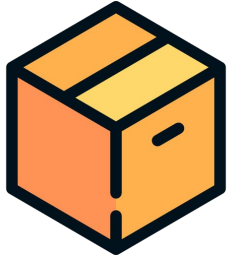
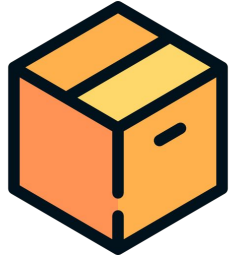
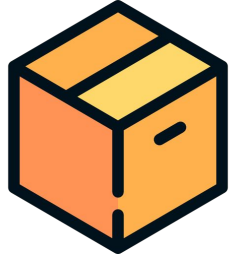
- 귀류법으로 증명 가능

Greedy Algorithm - 예시

* 어쩌서 이렇게 푸는가?

- 실제 문제에서 이 하나하나의 과정들은 그렇게 간단하지 않다
- 그리디 문제의 난이도는 대부분 그리디임을 알아내고 최적 전략을 짜는 데에서 나옴
- 구현이 까다로운 문제는 있을지언정 복잡한 자료구조를 응용하는 그리디는 거의 없다
- 문제에서 어떻게 나타나는지 기초적인 예제들을 통해 이해해 보자
- 대부분의 문제에서 이런 과정들을 종합적으로 생각해야 한다

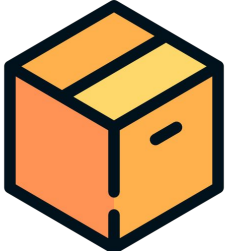
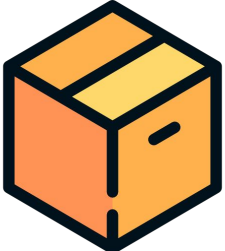
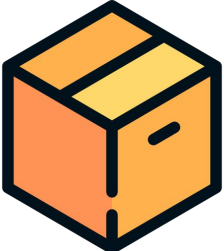
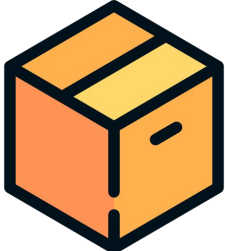
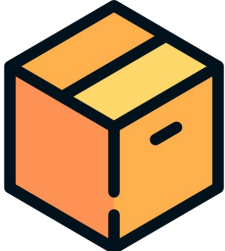
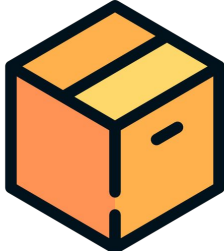
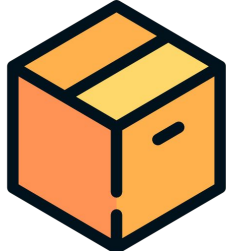
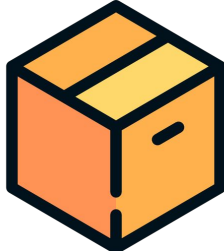
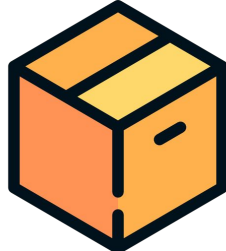
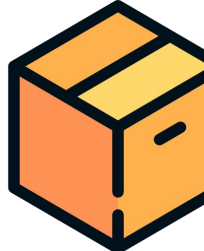
Greedy Algorithm - 한계

										
상자 번호	1	2	3	4	5	6	7	8	9	10
상자 가치	3	12	7	5	2	23	19	18	3	15

* 상자들 중 2개를 골라서 그 가치 합이 60이 되게 하려면?

- 가치가 60이랑 가장 가까운 것부터 고르면 5가 골라지는데 1은 없다

Greedy Algorithm - 한계

										
상자 번호	1	2	3	4	5	6	7	8	9	10
상자 가치	3	12	7	5	2	23	19	18	3	15

* 상자들 중 2개를 골라서 그 가치 합이 6이 되게 하려면?

- 가치가 6이랑 가장 가까운 것부터 고르면 5가 골라지는데 1은 없다
- 하지만 문제의 해는 분명 존재함
- set을 이용하는 등 여러 가지 방법으로 풀 수 있지만 그리디는 안됨
- 그리디로 이 문제를 풀 수 있는지 알아내는 안목을 길러야 -> 문제를 많이

풀기

처리 순서 고려 - 1931. 회의실 배정

* 문제

문제

한 개의 회의실이 있는데 이를 사용하고자 하는 N 개의 회의에 대하여 회의실 사용표를 만들려고 한다. 각 회의 i 에 대해 시작시간과 끝나는 시간이 주어져 있고, 각 회의가 겹치지 않게 하면서 회의실을 사용할 수 있는 회의의 최대 개수를 찾아보자. 단, 회의는 한번 시작하면 중간에 중단될 수 없으며 한 회의가 끝나는 것과 동시에 다음 회의가 시작될 수 있다. 회의의 시작시간과 끝나는 시간이 같을 수도 있다. 이 경우에는 시작하자마자 끝나는 것으로 생각하면 된다.

입력

첫째 줄에 회의의 수 N ($1 \leq N \leq 100,000$)이 주어진다. 둘째 줄부터 $N+1$ 줄까지 각 회의의 정보가 주어지는데 이것은 공백을 사이에 두고 회의의 시작시간과 끝나는 시간이 주어진다. 시작 시간과 끝나는 시간은 $2^{31}-1$ 보다 작거나 같은 자연수 또는 0이다.

출력

첫째 줄에 최대 사용할 수 있는 회의의 최대 개수를 출력한다.

처리 순서 고려 - 1931. 회의실 배정

* 최적의 전략은 무엇인가?

- 끝나는 시간이 빠른 회의부터 진행하는 것이 최적
- 귀류법을 통해서 위의 전략이 최적임을 증명할 수 있다.
- 그런데 문제를 처음 보았을 때 어떻게 '끝나는 시간이 빠른 것부터 진행하면 좋다'는 것을 알 수 있을까?

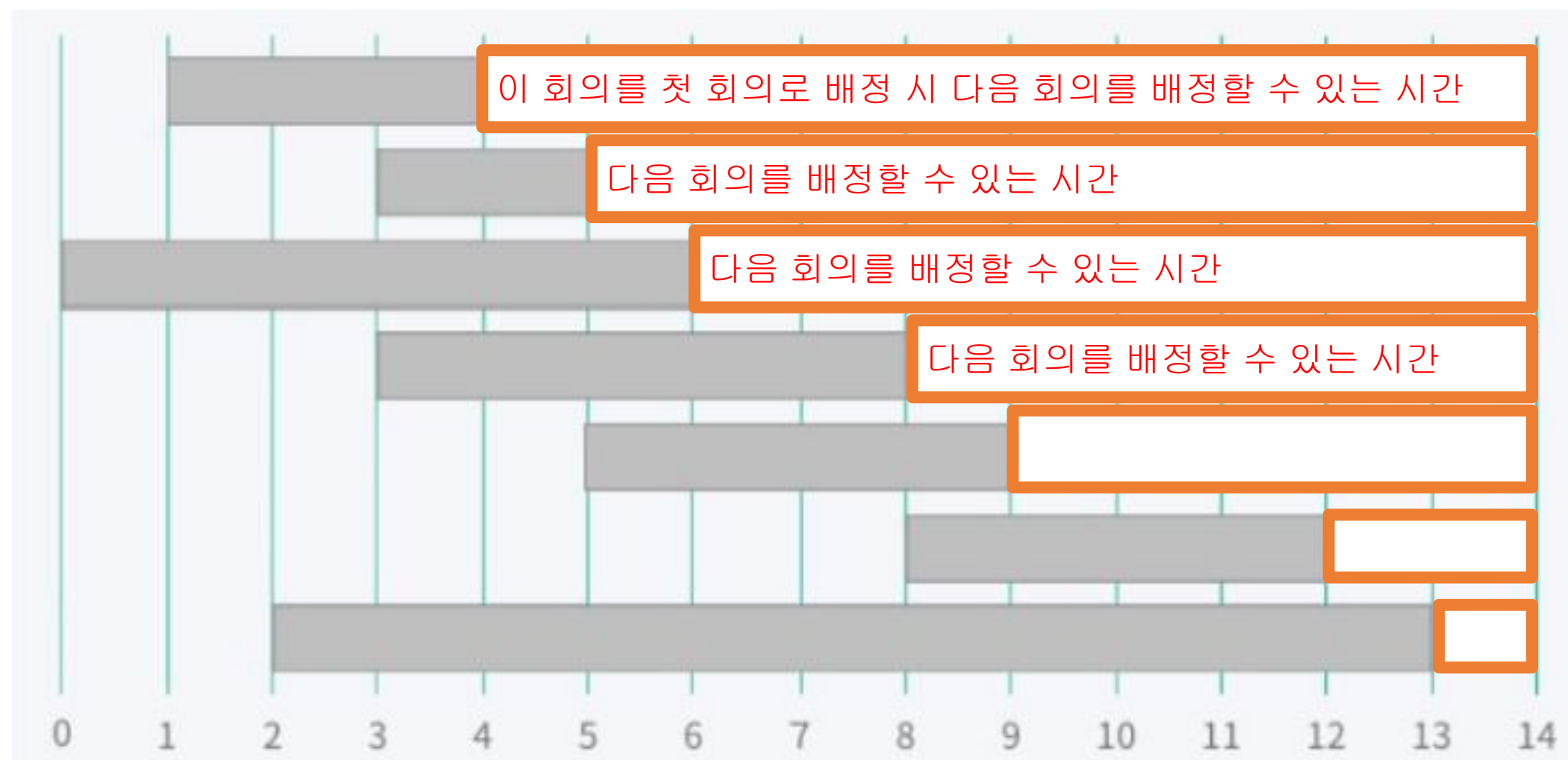
* 회의가 진행되고 있지 않은 여유 시간이 많으면 좋다

- k개 회의를 회의실에 배정했을 때, 다음 후보는 1~k번 회의가 진행되고 있지 않은 시간에 배정되어야 한다
- 회의를 배정할 때마다, 회의가 배정되어 있지 않음이 보장된 시간이 길수록 좋다
- 회의가 끝나는 시간이 빠르다는 것은 그 회의를 배정했을 때 '회의가 배정되어 있지 않음이 보장된 시간'이 더 길다는 뜻

처리 순서 고려 - 1931. 회의실 배정

* 회의가 진행되고 있지 않은 여유 시간이 많으면 좋다

- 회의가 끝나는 시간이 빠르다는 것은 그 회의를 배정했을 때 ‘회의가 배정되어 있지 않음’이 보장된 시간’이 더 길다는 뜻



처리 순서 고려 - 1931. 회의실 배정

* 회의가 진행되고 있지 않은 여유 시간이 많으면 좋다

- 회의가 끝나는 시간이 빠르다는 것은 그 회의를 배정했을 때 ‘회의가 배정되어 있지 않음’이 보장된 시간’이 더 길다는 뜻
- 따라서 현재 배정 가능한 회의 중 끝나는 시간이 가장 빠른 회의를 고르는 것이 부분 문제의 최적 전략
- 그 말은 다음 회의를 고를 때 회의를 배정할 수 있는 시간 선택지를 최대한 늘리는 것과 같다
- 따라서 끝나는 시간이 빠른 순서대로 정렬 후 그 순서대로 회의를 고르면 된다
- 단 이전 회의들과 시간이 겹치면 안 된다는 제약에 유의하여 구현(여기서도 제약을 생각해야 한다)

2022 Winter Algorithm Camp

처리 순서 고려 - 연습문제

* 필수문제

11399. ATM (Silver 3)

* 연습문제

9237. 이장님 초대(Silver 5)

16206. 롤케이크(Silver 1)

6068. 시간 관리하기 (Gold 5)

14908. 구두 수선공(Gold 2)

제약에 대한 고려 - 23559. 밥

* 문제

제주대 학생회관 식당에는 두 개의 메뉴가 있다. 코너 A로 가면 5,000원짜리 메뉴를 먹을 수 있고, 코너 B로 가면 1,000원짜리 메뉴를 먹을 수 있다.

준원이는 대면 수업이 시작되는 바람에 이제 남은 학기의 N 일동안 매일 학식의 두 메뉴 중 정확히 하나를 골라서 먹어야 한다. N 일간의 두 메뉴는 이미 공지되어 있고, 준원이는 이미 모든 날의 각 메뉴가 얼마나 맛있을지 수치를 매겨 두었다.

준원이는 N 일간 학식에 총 X 원 이하를 써야 한다.

여러분이 N 일간 준원이의 메뉴를 잘 골라서, 고른 메뉴의 맛의 합을 최대화 해주자!

입력

첫째 줄에는 두 정수 N , X 가 주어진다.

둘째 줄부터 N 개의 줄에, 각 날에 먹을 수 있는 5,000원짜리 메뉴의 맛 A 와 1,000원짜리 메뉴의 맛 B 가 공백을 사이에 두고 주어진다.

제약에 대한 고려 - 23559. 밥

* 문제

입력

첫째 줄에는 두 정수 N , X 가 주어진다.

둘째 줄부터 N 개의 줄에, 각 날에 먹을 수 있는 5,000원짜리 메뉴의 맛 A 와 1,000원짜리 메뉴의 맛 B 가 공백을 사이에 두고 주어진다.

출력

준원이가 고른 메뉴들의 맛의 합을 최대화했을 때의 값을 출력하라.

제한

- $1 \leq N \leq 100\,000$
- $1\,000N \leq X \leq 5\,000N$
- $1 \leq A \leq 10,000, 1 \leq B \leq 10,000$

제약에 대한 고려 - 23559. 밥

* 최적의 전략은 무엇인가?

- 가능하다면 모든 날에 5000원짜리와 1000원짜리 중 더 맛 수치가 높은 것을 고르는 것이 이득이다

* 고려할 제약 사항은 무엇인가?

- 예산에 제한이 있어서 5000원짜리가 더 맛 수치가 높더라도 고르지 못하는 상황이 올 수 있다
- 즉 5000원짜리 학식을 먹을 수 있는 날짜의 수가 제한되어 있다
- 1000원짜리를 먹는 것이 더 이득일 수도 있다

* 그럼 어떤 것부터 골라야 하는가?

- 어차피 1000원짜리는 늘 먹을 수 있다
- 따라서 5000원짜리가 1000원짜리보다 맛 수치가 높으며 1000원짜리와 차이가 큰 것을 골라야

제약에 대한 고려 - 23559. 밥

* 이를 어떻게 구현하는가?

- 5000원짜리와 1000원짜리의 차이를 생각한다
- i 번째 날의 5000원짜리 메뉴의 맛을 $A[i]$ 라 하고 1000원짜리 메뉴의 맛을 $B[i]$ 라 할 때 $A[i]-B[i]$ 가 음수인 날은 1000원짜리를 먹는 게 더 이득이므로 5000원짜리를 고르면 안 된다
- $A[i]-B[i]$ 가 양수라도 예산이 초과되면 고르지 못한다
- $A[i]-B[i]$ 순서대로 정렬하고 예산의 제약을 생각하면서 순서대로 처리

2022 Winter Algorithm Camp

제약에 대한 고려 - 연습문제

* 필수문제

20921. 그릇고 그런 사이
(Silver 2)

* 연습문제

11047. 동전 0 (Silver 2)
1946. 신입 사원 (Silver 1)

최적 전략을 반복하는 그리디

* 정렬로 모든 그리디를 처리할 수 있는가?

- 당장 이득이 되는 것부터 처리하는 것만 그리디 알고리즘이라고 생각한다면 정렬이 거의 필수적이다
- 그러나 처음에 언급했듯 그리디는 부분적인 최적 전략을 반복하는 알고리즘을 일반적으로 칭하는 용어
- 원소들을 정렬 후 하나씩 처리하는 것 외에도 다른 최적전략을 취하는 그리디 문제도 많다
- 그리디라고 직접적으로 생각하고 푸는 경우는 적지만 꽤 많이 나오는 문제 유형

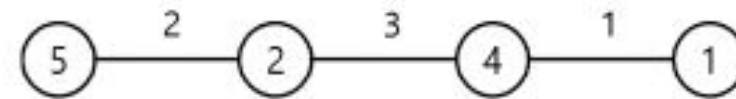
최적 전략의 반복 - 13305. 주유소

* 문제

어떤 나라에 N개의 도시가 있다. 이 도시들은 일직선 도로 위에 있다. 편의상 일직선을 수평 방향으로 두자. 제일 왼쪽의 도시에서 제일 오른쪽의 도시로 자동차를 이용하여 이동하려고 한다. 인접한 두 도시 사이의 도로들은 서로 길이가 다를 수 있다. 도로 길이의 단위는 km를 사용한다.

처음 출발할 때 자동차에는 기름이 없어서 주유소에서 기름을 넣고 출발하여야 한다. 기름통의 크기는 무제한이어서 얼마든지 많은 기름을 넣을 수 있다. 도로를 이용하여 이동할 때 1km마다 1리터의 기름을 사용한다. 각 도시에는 단 하나의 주유소가 있으며, 도시마다 주유소의 리터당 가격은 다를 수 있다. 가격의 단위는 원을 사용한다.

예를 들어, 이 나라에 다음 그림처럼 4개의 도시가 있다고 하자. 원 안에 있는 숫자는 그 도시에 있는 주유소의 리터당 가격이다. 도로 위에 있는 숫자는 도로의 길이를 표시한 것이다.



제일 왼쪽 도시에서 6리터의 기름을 넣고, 더 이상의 주유 없이 제일 오른쪽 도시까지 이동하면 총 비용은 30원이다. 만약 제일 왼쪽 도시에서 2리터의 기름을 넣고($2 \times 5 = 10$ 원) 다음 번 도시까지 이동한 후 3리터의 기름을 넣고($3 \times 2 = 6$ 원) 다음 도시에서 1리터의 기름을 넣어($1 \times 4 = 4$ 원) 제일 오른쪽 도시로 이동하면, 총 비용은 20원이다. 또 다른 방법으로 제일 왼쪽 도시에서 2리터의 기름을 넣고($2 \times 5 = 10$ 원) 다음 번 도시까지 이동한 후 4리터의 기름을 넣고($4 \times 2 = 8$ 원) 제일 오른쪽 도시까지 이동하면, 총 비용은 18원이다.

각 도시에 있는 주유소의 기름 가격과, 각 도시를 연결하는 도로의 길이를 입력으로 받아 제일 왼쪽 도시에서 제일 오른쪽 도시로 이동하는 최소의 비용을 계산하는 프로그램을 작성하시오.

최적 전략의 반복 - 13305. 주유소

* 문제

입력

표준 입력으로 다음 정보가 주어진다. 첫 번째 줄에는 도시의 개수를 나타내는 정수 N ($2 \leq N \leq 100,000$)이 주어진다. 다음 줄에는 인접한 두 도시를 연결하는 도로의 길이가 제일 왼쪽 도로부터 $N-1$ 개의 자연수로 주어진다. 다음 줄에는 주유소의 리터당 가격이 제일 왼쪽 도시부터 순서대로 N 개의 자연수로 주어진다. 제일 왼쪽 도시부터 제일 오른쪽 도시까지의 거리는 1 이상 1,000,000,000 이하의 자연수이다. 리터당 가격은 1 이상 1,000,000,000 이하의 자연수이다.

출력

표준 출력으로 제일 왼쪽 도시에서 제일 오른쪽 도시로 가는 최소 비용을 출력한다.

최적 전략의 반복 - 13305. 주유소

* 어떤 주유소에서 기름을 넣고 가는 게 이득인가?

- 리터당 가격이 가장 싼 주유소가 있는 도시에서 최대한 많은 기름을 넣는 게 이득이다
- 그런데 도시는 특정 순서대로만 방문할 수 있음
- 따라서 단순히 리터당 가격으로 정렬 후 순서대로 처리하는 건 쉽지 않음

최적 전략의 반복 - 13305. 주유소

* 주유소에서 기름을 넣어야 하는 도시는 어떻게 정하는가?

- i 번째 도시가 있을 때 $(i-1)$ 번째에서 i 번째 도시로 가는 거리에서 소모되는 기름은 $1 \sim (i-1)$ 번째 도시의 주유소에서 넣을 수 있음
- 따라서 $(i-1)$ 번째에서 i 번째 도시로 가는 거리에 소모되는 기름은 $1 \sim (i-1)$ 번째 도시의 주유소들 중 가장 리터당 가격이 싼 도시의 주유소에서 넣는다고 생각할 수 있다

* 최적 전략과 시행 방법은?

- 임의의 $(i-1)$ 번 도시에서 i 번 도시로 가는 기름은 $1 \sim (i-1)$ 번 도시의 주유소들 중 가장 리터당 가격이 싼 주유소에서 넣고 오는 게 최적 전략
- 이 전략을 1번부터 n 번 도시에 대해서 순서대로 반복하면서 주유 비용의 총합을 구하면 된다

2022 Winter Algorithm Camp

최적 전략의 반복 - 연습문제

* 연습문제

13413. 오셀로 재배치 (Silver 4)

19941. 햄버거 분배 (Silver 3)

11501. 주식 (Silver 2)

12931. 두 배 더하기 (Gold 4)

Greedy Algorithm - 발상

* 그리디 문제를 어떻게 풀 것인가?

- 문제의 상황이 그리디 알고리즘에 적합한지를 살핀다
- 가장 크거나, 작거나, 길거나, 어떤 함수의 결과가 덜하거나 등 어떤 기준으로 순서를 세워 그 순서대로 처리하면 이득이 되는지를 생각
- 직관 혹은 증명을 통해 어떤 기준이 확실하다면 구현
- 만약 그 기준이 보이지 않는다면 **DP** 등 다른 풀이를 생각해 보자
- 접해 보지 못한 유형이라면 그리디임을 떠올리기가 쉽지는 않다