

# 탐욕 알고리즘 (Greedy Algorithms)

- 01 탐욕 알고리즘 이론
- 02 탐욕 알고리즘 예시 문제 풀이

신 제 용

# 01 탐욕 알고리즘 이론

당장 좋은 것부터 하나씩 해 나가는 탐욕 알고리즘을 배워봅니다.

학습 키워드 – 탐욕 알고리즘, 그리디 알고리즘, Greedy

Chapter 01

탐욕 알고리즘 이론

# 탐욕 알고리즘 (Greedy algorithms)

- 매 순간 현재 기준으로 최선의 답을 선택해 나가는 기법
  - 빠르게 근사치를 계산할 수 있다.
  - 결과적으로는 최적해가 아닐 수도 있다.
- 탐욕 알고리즘의 결과가 최적해인 경우를 알아야 한다.

Chapter 01

탐욕 알고리즘 이론

# 활동 선택 문제 (Activity selection problem)

- N 개의 활동과 각 활동의 시작/종료 시간이 주어졌을 때,  
한 사람이 최대한 많이 할 수 있는 활동의 수 구하기

Activity	A	B	C	D	E
시작	1	4	2	4	6
종료	5	5	3	7	10

Chapter 01

탐욕 알고리즘 이론

# 활동 선택 문제 (Activity selection problem)

- 활동을 종료 시간 기준으로 오름차순 정렬
- 먼저 종료되는 활동부터 그리디하게 선택  
→ 문제 조건에 따라 시간이 겹치는 경우는 제외

Activity	C	A	B	D	E
시작	2	1	4	4	6
종료	3	5	5	7	10

Chapter 01

탐욕 알고리즘 이론

# 거스름돈 문제

- 동전의 개수가 가장 적게 거스름돈을 돌려주는 문제
  - 잔돈 890원
  - 동전의 종류: 10, 50, 100, 500
  - 큰 동전부터 개수를 계산

잔돈	500	100	50	10
개수	1	3	1	4

Chapter 01

탐욕 알고리즘 이론

# 거스름돈 문제

- 동전의 종류가 배수로 증가하지 않을 경우, 그리디 알고리즘이 최적해가 아닐 수 있다.
  - 동전의 종류: 10, 50, 100, 400, 500
  - 동적계획법(DP)에서 해결 방법을 배웁니다!

그리디 알고리즘 결과

잔돈	500	400	100	50	10
개수	1	0	3	1	4

우리가 원하는 정답

잔돈	500	400	100	50	10
개수	0	2	0	1	4

Chapter 01

탐욕 알고리즘 이론

# 그리디 알고리즘 최적해 조건

- 그리디 알고리즘은 빠르지만 최적해를 보장하지는 못함
- 하기 두 가지 조건에 해당하는 경우 적용 가능
  - 탐욕적 선택 특성 (Greedy choice property)  
지금 선택이 다음 선택에 영향을 주지 않음
  - 최적 부분 구조 (Optimal substructure)  
전체 문제의 최적해는 부분 문제의 최적해로 이루어짐

Chapter 01

탐욕 알고리즘 이론



# 02 탐욕 알고리즘 예시 문제 풀이

다음 챕터에서는 탐욕 알고리즘이 적용 가능한  
예시 문항을 학습합니다.

Chapter 01

탐욕 알고리즘 이론

# 02 탐욕 알고리즘 예시 문제 풀이

탐욕 알고리즘은 단순하고 간단하지만, 언제 적용 가능한지 아는 것이 핵심입니다. 탐욕 알고리즘이 적용 가능한 상황을 예시 문제를 통해 알아봅니다.

학습 키워드 – 탐욕 알고리즘, 최적해, 구현

## Chapter 02

### 탐욕 알고리즘 예시 문제 풀이

# Problem1

---

## 문제 설명

---

활동 선택 문제를 해결하세요!

각 활동이 `activity[i] = [시작시간, 끝나는시간]` 으로 주어질 때,

시간이 겹치지 않도록 선택할 수 있는 가장 많은 활동의 수를 구하시오.

## 매개변수 형식

---

```
activity = [[1, 5], [4, 6], [3, 5], [7, 12], [5, 11], [13, 15]]
```

## 반환값 형식

---

3

## Chapter 02

탐욕 알고리즘 예시  
문제 풀이

# Problem2

## 문제 설명

양의 정수가 담긴 문자열 `s`가 있다고 하자. 이 문자열에서 `k`개의 숫자를 제거해, 가장 작은 숫자를 만들고자 한다.

이렇게 만든 가장 작은 숫자를 담은 문자열을 출력하시오.

단, `k`개의 문자열을 제거한 결과는 앞에 불필요한 `0`이 포함될 수 있으며,

최종 출력에는 이 불필요한 `0`는 제거하여 출력하시오. (예시 입출력 참고)

## 입력 예시

```
s = "105990"
```

```
k = 1
```

## 출력 예시

```
"5990"
```

## 입출력 설명

맨 앞의 `'1'`을 제거하면 `"05990"`이 된다. 불필요한 `'0'`를 제거한 최종 출력은 `"5990"`이다.

## Chapter 02

### 탐욕 알고리즘 예시 문제 풀이

# Problem3

---

## 문제 설명

---

숫자 `num` 이 주어졌을 때, 최대 한번의 자릿수 교환을 통해 최대의 숫자를 만들어 내려고 한다.

즉, 자릿수를 교환하지 않았을 때가 더 큰 숫자인 경우, 원래 숫자를 그대로 출력해야 한다.

위 프로그램을 작성하시오.

## 매개변수 형식

---

```
num = 43824
```

## 반환값 형식

---

```
83424
```

## 입출력 예시 설명

---

0번째 인덱스의 숫자 `8` 과 2번째 인덱스의 숫자 `4` 의 자릿수를 바꾸었을 때 가장 큰 수가 된다.

## Chapter 02

### 탐욕 알고리즘 예시 문제 풀이