



# Greedy

미래를 생각하지 않고 각 단계에서 가장 최선의 선택을 하는 기법

문제를 해결하는 과정에서 매 순간, 최적이라 생각되는 해답 (locally optimal solution)을 찾으며, 이를 토대로 최종 문제의 해답 (globally optimal solution)에 도달하는 문제 해결 방식

## Greedy Algorithm 문제를 해결하는 방법

- 선택 절차 (Selection Procedure): 현재 상태에서의 최적의 해답을 선택한다
- 적절성 검사 (Feasibility Check): 선택된 해가 문제의 조건을 만족하는지 검사한다.
- 해답 검사 (Solution Check): 원래의 문제가 해결되었는지 검사하고, 해결되지 않았다면 선택 절차로 돌아가 위의 과정을 반복한다.

## Greedy Algorithm을 적용하기 위한 조건 2가지

- 탐욕스런 선택 속성 (Greedy Choice Property): 앞의 선택이 이후의 선택에 영향을 주지 않는다
- 최적 부분 구조 (Optimal Substructure): 문제에 대한 최종 해결 방법은 부분 문제에 대한 최적 문제 해결 방법으로 구성된다

(DP의 경우 문제가 overlapping 되므로 다음에 풀 문제가 이전의 작은 문제의 결과에 영향을 받게 된다. 반면 그리디 알고리즘은 다른 경우의 결과에 상관없이 최적해를 구하기 때문에 이전의 경우에 영향을 받아서는 안된다.)

- 탐욕 알고리즘은 위 조건이 성립하지 않는 경우는 최적의 결과를 도출하지 못하지만, 어느 정도 최적에 근사한 값을 빠르게 도출할 수 있는 장점이 있다. 이 장점으로 인해 탐욕 알고리즘은 근사 알고리즘으로 사용할 수 있다.
- 탐욕 알고리즘을 적용해도 언제나 최적해를 구할 수 있는 문제(**매트로이드**)가 있고, 이러한 문제에 탐욕 알고리즘을 사용해서 빠른 계산 속도로 답을 구할 수 있어서 실용적으로 사용할 수 있다.
- 코딩 테스트에서의 대부분의 그리디 문제는 탐욕법으로 얻은 해가 최적의 해가 되는 상황을 추론할 수 있어야 풀리도록 출제된다.

## 근사 알고리즘(Approximation Algorithm)이란?

어떤 최적화 문제에 대한 해의 근사값을 구하는 알고리즘

이 알고리즘은 가장 최적화되는 답을 구할 수는 없지만, 비교적 빠른 시간에 계산이 가능하며(polynomial 시간안에 해결) 어느 정도 보장된 근사해를 계산할 수 있다.

### ! 예시 1 – 매트로이드

하나몬은 오늘도 편의점에서 열심히 아르바이트하고 있다. 손님으로 온 박해커는 과자와 음료를 하나씩 집어 들었고, 물건 가격은 총 4,040원이 나왔다. 박해커는 계산을 하기 위해 5,000원을 내밀며, 거스름돈은 동전의 개수를 최소한으로 하여 거슬러 달라고 하였다.

### 👉 이때 하나몬은 어떻게 거슬러 주어야 할까?

탐욕 알고리즘으로 동전의 개수를 헤아리는 일은 일반적으로 거스름돈으로 동전을 선택하는 방법과 동일하다. 거스름돈 960원을 채우기 위해서 먼저 500원짜리 동전을 한 개를 선택한다. 그다음은 100원짜리 동전을 네 개 선택하고, 그다음엔 50원짜리 동전과 10원짜리 동전을 각각 하나씩 선택할 것이다.

### 👉 탐욕 알고리즘의 문제 해결 과정을 적용

#### 선택 절차

- 거스름돈의 동전 개수를 줄이기 위해 현재 가장 가치가 높은 동전을 우선 선택한다.

#### 적절성 검사

- 1번 과정을 통해 선택된 동전들의 합이 거슬러 줄 금액을 초과하는지 검사한다.
- 초과하면 가장 마지막에 선택한 동전을 삭제하고, 1번으로 돌아가 한 단계 작은 동전을 선택한다.

## 해답 검사

- 선택된 동전들의 합이 거슬러 줄 금액과 일치하는지 검사한다.
- 액수가 부족하면 1번 과정부터 다시 반복한다.

## 👉 이 과정을 통해 얻은 문제에 대한 해답

가장 가치가 높은 동전인 500원 1개를 먼저 거슬러 주고 잔액을 확인한 뒤, 이후 100원 4개, 50원 1개, 10원 1개의 순서대로 거슬러 준다.

```
N = 960
cnt = 0
for coin in [500,100,50,10]:
    cnt += N//coin
    N %= coin
print(cnt)
```

## Reference)

[알고리즘] 탐욕 알고리즘(Greedy Algorithm) - 하나몬

탐욕 알고리즘(Greedy Algorithm)이란? Greedy는 ‘탐욕스러운, 욕심 많은’ 이란 뜻이다. 탐욕 알고리즘은 말 그대로 선택의 순간마다 당장 눈앞에 보이는 최적의 상황만을 쫓아 최종적인 해답에 도달하는

 <https://hanamon.kr/알고리즘-탐욕알고리즘-greedy-algorithm/>

