

chapter8. 탐욕 알고리즘

- Hello Coding 그림으로 개념을 이해하는 알고리즘 -



TABLE OF CONTENTS



1. 탐욕 알고리즘
2. 탐욕 알고리즘 예제(동전문제)
3. 탐욕 알고리즘의 한계

1. 탐욕 알고리즘

그리디 알고리즘 or 탐욕 알고리즘이라고 부른다.

최적의 해에 가까운 값을 구하기 위해 사용된다.

기술적인 용어로 말하자면 각 단계에서 국소 최적해를 찾음으로써 최종적으로는 전역 최적해를 구하게 된다고 합니다.

여러 경우 중 하나를 구해야할 때마다, 매 순간 최적이라고 생각하는 경우를 선택하는 방식으로 진행해서 최종적인 값을 구하는 방식

탐욕 알고리즘이 항상 정확한 정답을 구할 수 있지는 않습니다...(그래도 근사값은 구할수있음)

2. 탐욕 알고리즘 예제 (동전문제)

Q. 지불해야 하는 값이 4720원일 때 1원, 50원, 100원, 500원 동전으로 동전의 수가 가장 적게 지불하시오.

- 가장 큰 동전부터 최대한 지불해야 하는 값을 채우는 방식으로 구현 가능
- 탐욕 알고리즘으로 매순간 최적이라고 생각하는 경우를 선택하면 됨

```
coin_list = [500,100,50,1]
```

```
def min_cost_count(value, coin_list):
```

```
    total_coin_count = 0
```

```
    details = []
```

```
    coin_list.sort(reverse=True)
```

```
    for coin in coin_list:
```

```
        coin_num = value // coin
```

```
        total_coin_count += coin_num
```

```
        value -= coin * coin_num
```

```
        details.append([coin, coin_num])
```

```
    return total_coin_count, details
```

```
print(min_cost_count(4720, coin_list))
```

```
(31, [[500, 9], [100, 2], [50, 0], [1, 20]])
```

coin_list.sort(reverse=True) → 리스트 원소들을 내림차순으로 정렬.

그냥 sort()하면 오름차순으로 정렬

// → 나누기 한 값에 소수점붙어있으면 땀.

참고

append는 한개의 인자만 넣을 수 있음.

extend는 여러개 넣을 수 있음.

insert는 원하는 자리에 넣을 수 있음.

3. 탐욕 알고리즘의 한계

1. 탐욕 알고리즘은 근사치 추정에 활용합니다.
2. 반드시 최적의 해를 구할 수 있지 않습니다. (정확도가 떨어짐)
3. 최적의 해에 가까운 값을 구하는 방법 중 하나입니다.