CodeCure/2021

Greedy Algorithm

- Easy -

안태진(taejin7824@gmail.com)
GitHub(https://github.com/Taejin1221)
소프트웨어학과 18학번
상명대학교 CodeCure 소프트부장

Contents

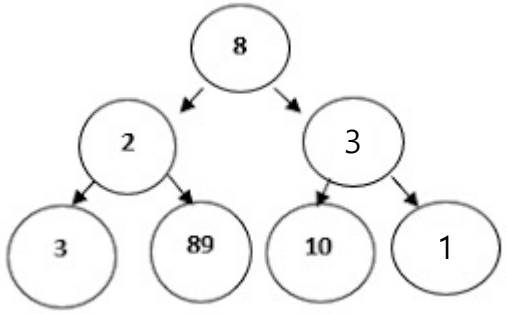
- Greedy Algorithm?
- Example
- 적용할 수 있는 문제 vs 적용할 수 없는 문제
- Problem

Greedy Algorithm?

- Greedy Algorithm (탐욕 알고리즘, 욕심쟁이 알고리즘)
  - 처음 배우는 생각하는? 수학적인? Algorithm!
    - 기업 코딩 테스트나 대회에서 많이 나옴! 중요!
  - 직관이 필요한 문제라 경험이 좀 필요 or 똑똑하면…
    - 직관: 이렇게 풀면 맞을 것 같은데?

  - 최종적인 답을 보기보단, 현재 가장 답같이 보이는 상태를 선택하며 진행해 나가는 알고리즘

  - 욕심쟁이처럼 지금 당장 좋은 것을 선택
    - 루트부터 리프까지 탐색하며 값을 더할 때 최소가 되는 값을 구하여라



Contents

- Greedy Algorithm?
- Example
- 적용할 수 있는 문제 vs 적용할 수 없는 문제
- Problem

Example

- 말로는 어렵다! 예제를 보자!

- Example 1
  - BOJ 1422: 숫자의 신
    - 주어진 숫자들로 n자리의 가장 큰 숫자를 만들어라
    - 대신 주어진 숫자들은 다 쓰자!

  - 해답
    - 그냥 가장 큰 숫자들을 앞에다 놓으면 된다! 그런 다음 가장 큰 것들을 앞에 반복하자
      - sort(reverse = True)
    - Greedy!

예제 입력 1 복사

```
3 3
3
2
7
```

예제 출력 1 복사

```
732
```

Example

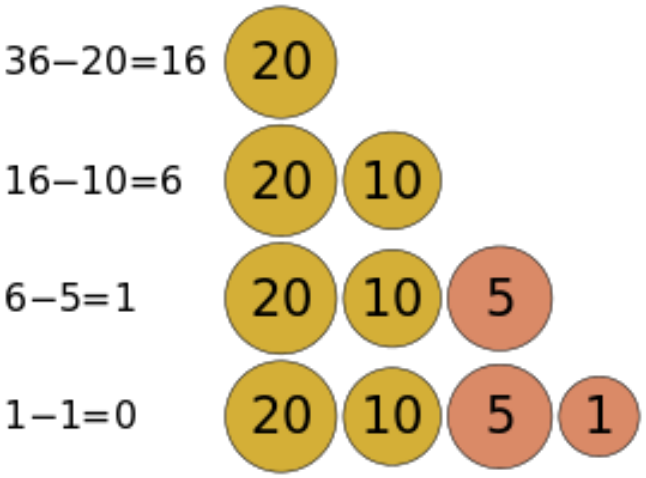
- Example 2
  - BOJ 11047: 동전 0
    - n개의 동전, k원이 주어질 때
    - k원을 만들기 위해 사용해야하는 동전의 최소 개수?
  - Solution
    - 최소 개수로 만드려면 가장 큰 것부터 때려 박자
      - 4,200 = 1,000 * 4 + 100 * 2 => 6개
    - Greedy!

예제 입력 1 복사

```
10 4200
1
5
10
50
100
500
1000
5000
10000
50000
```

예제 출력 1 복사

```
6
```



Contents

- Greedy Algorithm?
- Example
- 적용할 수 있는 문제 vs 적용할 수 없는 문제
- Problem

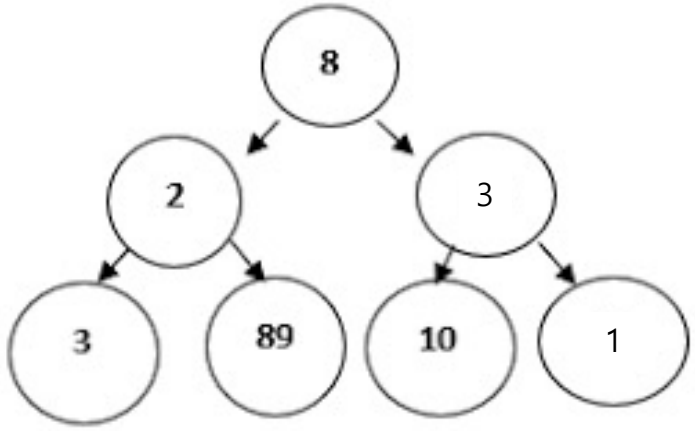
적용할 수 있는 문제 vs 적용할 수 없는 문제

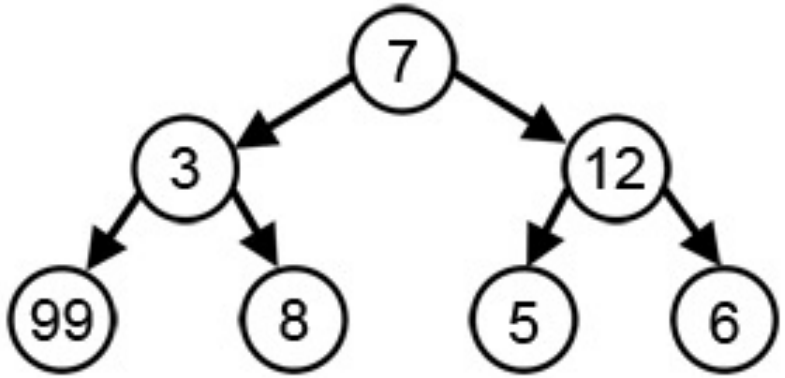
- Greedy를 적용할 수 있는 문제
  - 지금 상태에서 좋은 것을 선택하는 것이 최종 답을 위한 선택인 경우
  - 부분해가 최적해일 때

- Greedy를 적용할 수 없는 문제
  - 지금 상태에서 좋은 것을 선택하는 것이 최종 답이 아닌 경우
  - 부분해가 최적해가 아닐 때

적용할 수 있는 문제 vs 적용할 수 없는 문제

- Example 1 (적용할 수 없는 문제)
  - 최소 탐색 값 구하기
    - 현재 선택한 노드에서 가장 작은 값을 선택하는 것이 최적해가 아님
  - Not Greedy!
    - 완전 탐색이나 Dynamic Programming으로 해결





적용할 수 있는 문제 vs 적용할 수 없는 문제

- Example 2 (적용할 수 없는 문제)
  - BOJ 2294: 동전 2
    - 동전 0와 똑같은 문제
    - 단, 동전이 coin[i + 1]이 coin[i]의 배수가 아님
  - Solution
    - 똑같이 가장 큰 것을 배치?
      - 15 = 12 * 1 + 1 * 3 => 4개
      - 15 = 5 * 3 => 3개
    - 따라서 완전탐색 or Dynamic Programming으로 풀어야한다!
      - 이 문제는 DP

예제 입력 1 복사

```
3 15
1
5
12
```

예제 출력 1 복사

```
3
```

Contents

- Greedy Algorithm?
- Example
- 적용할 수 있는 문제 vs 적용할 수 없는 문제
- Problem

Problem

- 대표적인 Greedy 문제 1
  - BOJ 11399: ATM
    - ATM이 딱 1대 있고 각 사람이 돈을 인출하는 시간이 주어질 때
    - 대기 시간을 최소로 하라

예제 입력 1 복사

```
5
3 1 4 3 2
```

    - Solution
      - 오른쪽이 대기 시간일 때 오른쪽 순서로 쓴다면
        - $3 + (3 + 1) + (3 + 1 + 4) + (3 + 1 + 4 + 3) + (3 + 1 + 4 + 3 + 2) =$
        - $3\times5 + 1\times4 + 4\times3 + 3\times2 + 2\times1 = 39$
      - 대기 순서가 앞으로 갈 수록 더 큰 수가 곱해진다
        - (뒤의 대기자들은 그 사람을 무조건 기다려야하니까)
      - 이를 최소로 만들기 위해선? 앞으로 갈수록 작은 수 (숫자가 뒤로 갈 수록 커져야함)
      - 즉 오름차순 정렬 한 뒤 더해주자!

Problem

- 대표적인 Greedy 문제 2
  - BOJ 1931: 회의실 배정
    - 회의의 시작 시간과 종료 시간이 주어질 때
    겹치지 않도록 회의를 최대한 많이 잡아라

  - Solution
    - Greedy하게 그냥 잡아보자
      - 첫번째 회의 잡고 그 다음에 잡을 수 있는 가장 가까운 회의 잡자

    - 하지만 입력이 저렇게 주어질리 없다
      - 모든 입력에 대해서 최대한 많은 회의를 잡으려면 어떻게 해야할까?
      - 끝나는 시간을 기준으로 정렬하자!
      - 최대한 빨리 끝나는 회의를 잡으면 최대한 많이 잡을 수 있다!

예제 입력 1 복사

```
11
1 4
3 5
0 6
5 7
3 8
5 9
6 10
8 11
8 12
2 13
12 14
```

예제 출력 1 복사

```
4
```

Homework

- 과제
  - 제출 방법은 동일
  - 문제집
    - https://www.acmicpc.net/group/workbook/view/11714/35352

감사합니다!