JavaScript 탐욕법 알고리즘

2) 그리디 문제 풀이 ①

그리디 문제 풀이 | 코딩 테스트에서 자주 등장하는 탐욕법 알고리즘 이해하기

강사 나동빈

JavaScript 탐욕법 알고리즘

2) 그리디 문제 풀이 ①

혼자 힘으로 풀어보기

문제 제목: 동전 0

문제 난이도: ★☆☆☆☆

문제 유형: 그리디

추천 풀이 시간: 20분

문제 해결 아이디어

- 거스름 돈 문제와 동일한 아이디어로 해결할 수 있다.
- 화폐의 종류가 N개이고, K원을 만들어야 한다.
- 또한 각 화폐의 단위는 서로 **배수 관계**에 해당한다.
- 본 문제는 그리디 알고리즘을 이용해 문제를 해결할 수 있다.

동전 0: https://www.acmicpc.net/problem/11047

문제 해결 아이디어

JavaScript
탐욕법
그리디
문제 풀이

[문제 해결 아이디어] 단순히 가장 큰 화폐 단위부터 먼저 거슬러준다.

- 각 화폐의 단위는 서로 배수 관계에 해당한다.
- 예를 들어 화폐 4개가 있다면, 10, 50, 100, 500원과 같은 형식이 전형적이다.
- 가치가 큰 동전은 가치가 작은 동전들의 합으로 표현될 수 있다.

동전 0: https://www.acmicpc.net/problem/11047

JavaScript 탐욕법
그리디 문제 풀이

문제 해결 아이디어

- $N = 10$개의 화폐가 존재하며, $K = 4790$원을 만드는 경우를 고려해 보자.

1	500
5	1000
10	5000
50	10000
100	50000

동전 0: https://www.acmicpc.net/problem/11047

JavaScript 탐욕법
그리디 문제 풀이

문제 해결 아이디어

- 먼저, 가능한 모든 **화폐 단위**를 내림차순으로 정렬하여 고려해 보자.

50000	10000	5000	1000	500
100	50	10	5	1

동전 0: https://www.acmicpc.net/problem/11047

JavaScript 탐욕법
그리디 문제 풀이

문제 해결 아이디어

- 하나씩 화폐의 단위를 확인하여, 해당 화폐로 나눌 때의 몫을 더하면 된다.

[1단계] 50000원에 대하여 처리한다.

50000	10000	5000	1000	500
100	50	10	5	1

- **남은 돈**: 4790
- **사용한 화폐의 수**: 0

동전 0: https://www.acmicpc.net/problem/11047

JavaScript 탐욕법
그리디 문제 풀이

문제 해결 아이디어

- 하나씩 화폐의 단위를 확인하여, 해당 화폐로 나눌 때의 몫을 더하면 된다.

[2단계] 10000원에 대하여 처리한다.

50000	10000	5000	1000	500
100	50	10	5	1

- **남은 돈**: 4790
- **사용한 화폐의 수**: 0

동전 0: https://www.acmicpc.net/problem/11047

JavaScript 탐욕법
그리디 문제 풀이

문제 해결 아이디어

- 하나씩 화폐의 단위를 확인하여, 해당 화폐로 나눌 때의 몫을 더하면 된다.

[3단계] 5000원에 대하여 처리한다.

50000	10000	5000	1000	500
100	50	10	5	1

- **남은 돈**: 4790
- **사용한 화폐의 수**: 0

동전 0: https://www.acmicpc.net/problem/11047

JavaScript 탐욕법
그리디 문제 풀이

문제 해결 아이디어

- 하나씩 화폐의 단위를 확인하여, 해당 화폐로 나눌 때의 몫을 더하면 된다.

[4단계] 1000원에 대하여 처리한다.

50000	10000	5000	1000	500
100	50	10	5	1

- **남은 돈**: 790
- **사용한 화폐의 수**: 4

동전 0: https://www.acmicpc.net/problem/11047

JavaScript 탐욕법
그리디 문제 풀이

문제 해결 아이디어

- 하나씩 화폐의 단위를 확인하여, 해당 화폐로 나눌 때의 몫을 더하면 된다.

[5단계] 500원에 대하여 처리한다.

50000	10000	5000	1000	500
100	50	10	5	1

- **남은 돈**: 290
- **사용한 화폐의 수**: 5

동전 0: https://www.acmicpc.net/problem/11047

JavaScript 탐욕법
그리디 문제 풀이

문제 해결 아이디어

- 하나씩 화폐의 단위를 확인하여, 해당 화폐로 나눌 때의 몫을 더하면 된다.

[6단계] 100원에 대하여 처리한다.

50000	10000	5000	1000	500
100	50	10	5	1

- **남은 돈**: 90
- **사용한 화폐의 수**: 7

동전 0: https://www.acmicpc.net/problem/11047

JavaScript 탐욕법
그리디 문제 풀이

문제 해결 아이디어

- 하나씩 화폐의 단위를 확인하여, 해당 화폐로 나눌 때의 몫을 더하면 된다.

[7단계] 50원에 대하여 처리한다.

50000	10000	5000	1000	500
100	50	10	5	1

- **남은 돈**: 40
- **사용한 화폐의 수**: 8

동전 0: https://www.acmicpc.net/problem/11047

JavaScript 탐욕법
그리디 문제 풀이

문제 해결 아이디어

- 하나씩 화폐의 단위를 확인하여, 해당 화폐로 나눌 때의 몫을 더하면 된다.

[8단계] 10원에 대하여 처리한다.

50000	10000	5000	1000	500
100	50	10	5	1

- **남은 돈**: 0
- **사용한 화폐의 수**: 12

동전 0: https://www.acmicpc.net/problem/11047

JavaScript 탐욕법
그리디 문제 풀이

문제 해결 아이디어

- 하나씩 화폐의 단위를 확인하여, 해당 화폐로 나눌 때의 몫을 더하면 된다.

[9단계] 5원에 대하여 처리한다.

50000	10000	5000	1000	500
100	50	10	5	1

- **남은 돈**: 0
- **사용한 화폐의 수**: 12

동전 0: https://www.acmicpc.net/problem/11047

문제 해결 아이디어

- 하나씩 화폐의 단위를 확인하여, 해당 화폐로 나눌 때의 몫을 더하면 된다.

[10단계] 1원에 대하여 처리한다.

50000	10000	5000	1000	500
100	50	10	5	1

- **남은 돈**: 0
- **사용한 화폐의 수**: 12

동전 0: https://www.acmicpc.net/problem/11047

JavaScript 탐욕법
그리디 문제 풀이

정답 코드 예시

```
let fs = require('fs');
let input = fs.readFileSync('/dev/stdin').toString().split('\n');

let n = Number(input[0].split(' ')[0]); // 동전의 개수
let k = Number(input[0].split(' ')[1]); // 만들어야 할 금액

let arr = [];
// 전체 동전(화폐 단위) 데이터 입력
for (let i = 1; i <= n; i++) arr.push(Number(input[i]));

let cnt = 0;
// 가치가 큰 동전부터 확인
for (let i = n - 1; i >= 0; i--) {
  cnt += parseInt(k / arr[i]); // 해당 동전을 몇 개 사용해야 하는지
  k %= arr[i]; // 해당 동전으로 모두 거슬러 준 뒤 남은 금액
}

console.log(cnt);
```

ATM: https://www.acmicpc.net/problem/11399

JavaScript 탐욕법
그리디 문제 풀이

혼자 힘으로 풀어보기

문제 제목: ATM

문제 난이도: ★☆☆☆☆

문제 유형: 그리디

추천 풀이 시간: 20분

ATM: https://www.acmicpc.net/problem/11399

문제 해결 아이디어

JavaScript
탐욕법
그리디
문제 풀이

- 각 사람이 돈을 인출하는데 **필요한 시간의 합의 최솟값**을 계산한다.

[문제 해결 아이디어] 시간이 적게 소요되는 사람부터 처리할 때, 필요한 시간의 합이 최소가 된다.

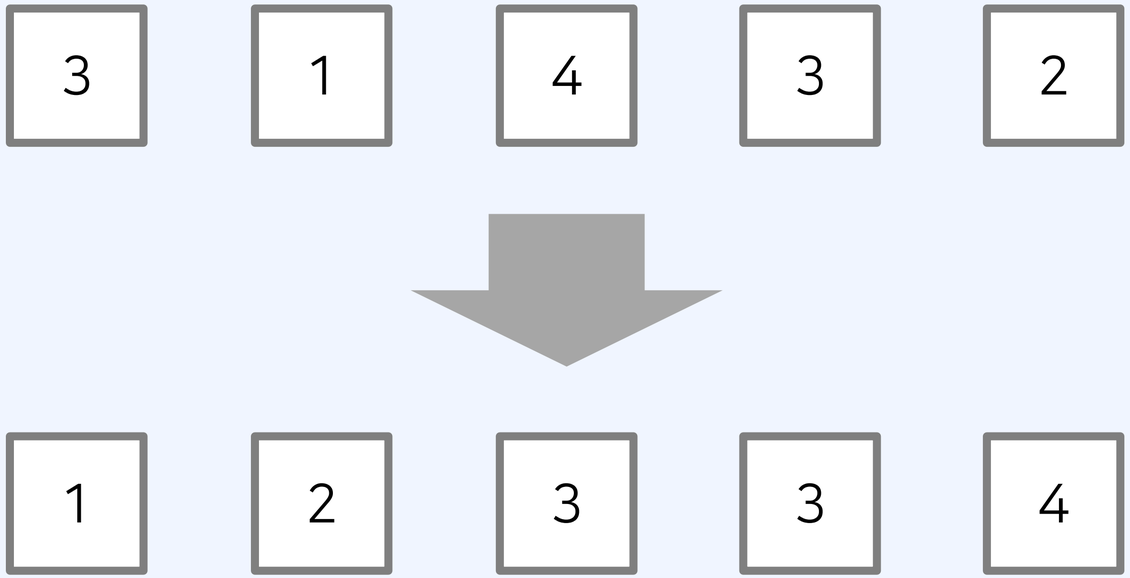
- 따라서 오름차순 정렬 이후에 누적 합을 계산하여 해결할 수 있다.

ATM: https://www.acmicpc.net/problem/11399

JavaScript 탐욕법
그리디 문제 풀이

문제 해결 아이디어

- 가장 먼저 오름차순 정렬을 수행한다.



ATM: https://www.acmicpc.net/problem/11399

문제 해결 아이디어

JavaScript
탐욕법
그리디
문제 풀이

[1단계]

- 현재 시간: 1
- 기다린 시간의 총 합: 1

ATM: https://www.acmicpc.net/problem/11399

문제 해결 아이디어

JavaScript 탐욕법
그리디 문제 풀이

[2단계]

- 현재 시간: 3
- 기다린 시간의 총 합: 1 + 3

ATM: https://www.acmicpc.net/problem/11399

문제 해결 아이디어

JavaScript
탐욕법
그리디
문제 풀이

[3단계]

- 현재 시간: 6
- 기다린 시간의 총 합: 1 + 3 + 6

ATM: https://www.acmicpc.net/problem/11399

문제 해결 아이디어

JavaScript
탐욕법
그리디
문제 풀이

[4단계]

- 현재 시간: 9
- 기다린 시간의 총 합: 1 + 3 + 6 + 9

ATM: https://www.acmicpc.net/problem/11399

문제 해결 아이디어

[5단계]

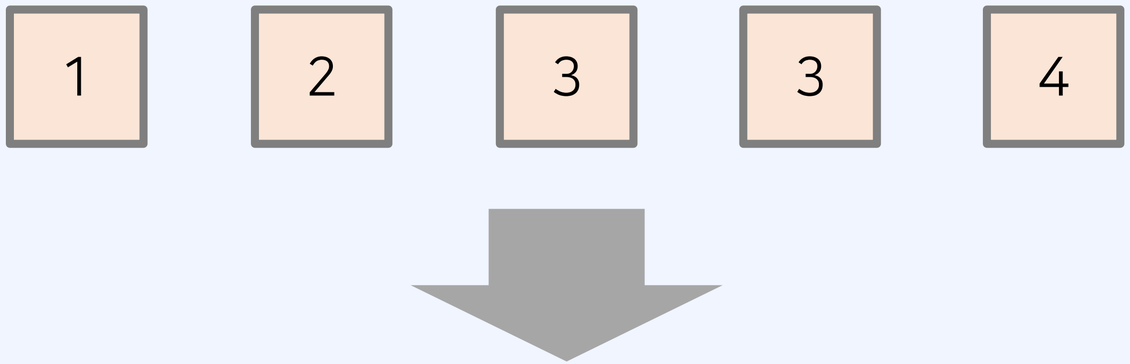
- 현재 시간: 13
- 기다린 시간의 총 합: 1 + 3 + 6 + 9 + 13

ATM: https://www.acmicpc.net/problem/11399

문제 해결 아이디어

[5단계]

- 현재 시간: 13
- 기다린 시간의 총 합: 1 + 3 + 6 + 9 + 13



- 따라서, 기다린 시간의 총 합은 1 + 3 + 6 + 9 + 13 = 32

ATM: https://www.acmicpc.net/problem/11399

JavaScript 탐욕법
그리디 문제 풀이

정답 코드 예시

```
let fs = require('fs');
let input = fs.readFileSync('/dev/stdin').toString().split('\n');

let n = Number(input[0]); // 사람의 수
let arr = input[1].split(' ').map(Number); // 모든 처리 시간 입력받기

// 오름차순 정렬
arr.sort((a, b) => a - b);

let answer = 0;
let summary = 0;
for (let i = 0; i < n; i++) {
  summary += arr[i]; // i번째 사람이 기다린 총 시간
  answer += summary; // 지금까지 소요된 총 시간
}

console.log(answer);
```

혼자 힘으로 풀어보기

JavaScript
탐욕법
그리디
문제 풀이

문제 제목: 읽어버린 괄호

문제 난이도: ★★☆☆☆

문제 유형: 그리디

추천 풀이 시간: 40분

잃어버린 괄호: https://www.acmicpc.net/problem/1541

문제 해결 아이디어

- 문제를 요약하자면 다음과 같다.
- 덧셈(+)과 뺄셈(-) 연산자로만 구성된 수식이 있을 때, 괄호를 적절히 넣어 값을 최소화한다.

[문제 해결 아이디어] 뺄셈(-) 연산자를 기준으로 최대한 많은 수를 묶는다.

읽어버린 괄호: https://www.acmicpc.net/problem/1541

문제 해결 아이디어

[예시 1]

- 현재 예시에서는 애초에 **마이너스(-)**가 존재하지 않으므로, 처리할 것이 없다.
- 10+20+30+40 = 100

읽어버린 괄호: https://www.acmicpc.net/problem/1541

문제 해결 아이디어

JavaScript
탐욕법
그리디
문제 풀이

[예시 2]

- 이 경우는 **마이너스(-)**가 존재한다.
- 어떻게 괄호를 묶어야, 가장 결과를 작게 만들 수 있을까?
- 55-50+40 → (55)-(50+40) = -35

잃어버린 괄호: https://www.acmicpc.net/problem/1541

문제 해결 아이디어

JavaScript
탐욕법
그리디
문제 풀이

[예시 3]

- 여러 개의 **마이너스(-)**가 존재하는 상황을 고려해 보자.
- 어떻게 괄호를 묶어야, 가장 결과를 작게 만들 수 있을까?
- 90+30-20+50-30+60-70+30+20

→ (90+30)-(20+50)-(30+60)-(70+30+20) = -160

읽어버린 괄호: https://www.acmicpc.net/problem/1541

JavaScript 탐욕법
그리디 문제 풀이

정답 코드 예시

```
let fs = require('fs');
let input = fs.readFileSync('/dev/stdin').toString().split('\n');

// 뺄셈(-) 연산자를 기준으로 나누어 여러 그룹 만들기
let groups = input[0].split('-');
let answer = 0;
for (let i = 0; i < groups.length; i++) {
  // 각 그룹 내부에서 덧셈(+) 연산 적용
  let cur = groups[i].split('+').map(Number).reduce((a, b) => a + b);
  if (i == 0) answer += cur; // 첫 번째 그룹은 항상 덧셈(+)
  else answer -= cur; // 두 번째 그룹부터 뺄셈(-)
}

console.log(answer);
```