JavaScript
이진 탐색 알고리즘
파라메트릭 서치 이해하기

파라메트릭 서치 이해하기 | 코딩 테스트에서 자주 등장하는 이진 탐색 알고리즘 이해하기

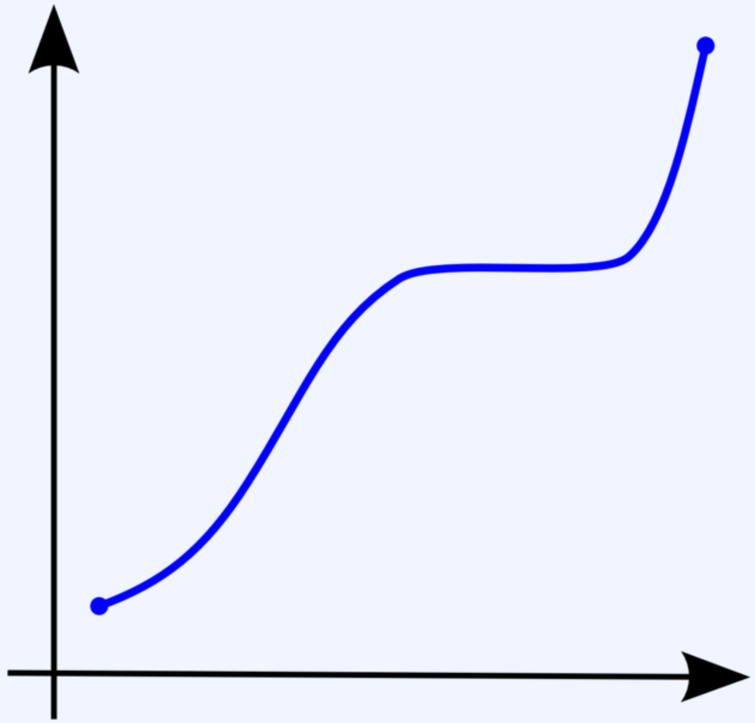
강사 나동빈

JavaScript
이진 탐색 알고리즘
파라메트릭 서치 이해하기

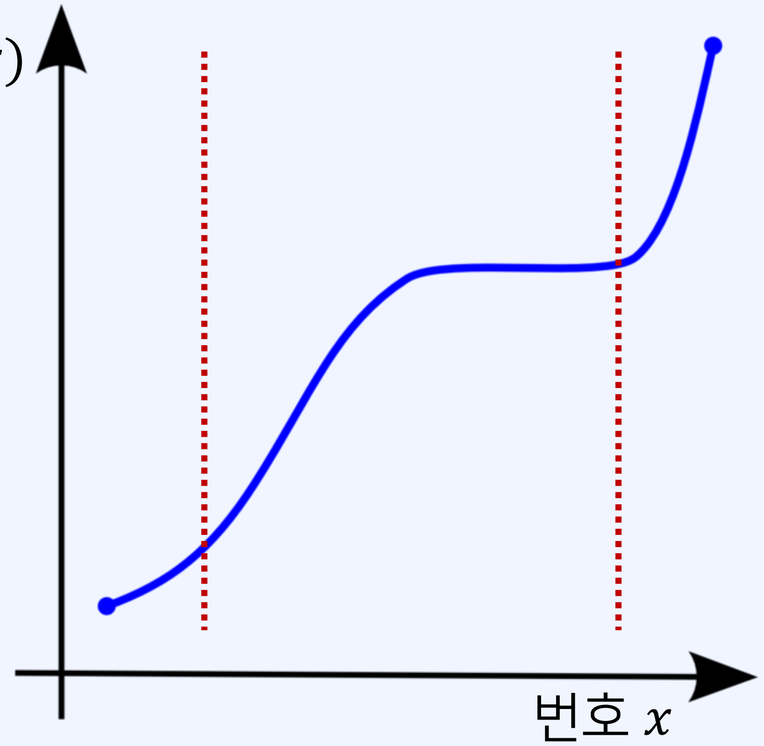
파라메트릭 서치 이해하기

- **이진 탐색 조건**: 변경할(최적화할) 값 x에 대하여 $f(x)$가 단조 증가 혹은 단조 감소
  - 단조 증가 함수: $x \leq y$이면 $f(x) \leq f(y)$인 경우
  - 일반적으로 조건(constraint)은 $f(x)$에 대하여 정의된다.

예시: 성적이 [a, b] 사이인 학생들 찾기



이진 탐색 활용 예시



파라메트릭 서치(Parametric Search)란?

- 최적화 문제를 결정 문제('예' 혹은 '아니오')로 바꾸어 해결하는 기법이다.
  - 예시: 특정한 조건을 만족하는 가장 알맞은 값을 빠르게 찾는 최적화 문제
- 일반적으로 코딩 테스트에서 파라메트릭 서치 문제는 **이진 탐색을 이용하여 해결**할 수 있다.
  - 파라메트릭 서치 문제의 목적 함수 예시:

$$\max_{x} x$$

$$s.t.\ f(x) \geq C$$

$$where\ f(x)\ is\ monotone\ decreasing\ function.$$

예산: https://www.acmicpc.net/problem/2512

혼자 힘으로 풀어보기

JavaScript
이진 탐색
파라메트릭
서치

문제 제목: 예산

문제 난이도: ★★☆☆☆

문제 유형: 이진 탐색, 파라메트릭 서치

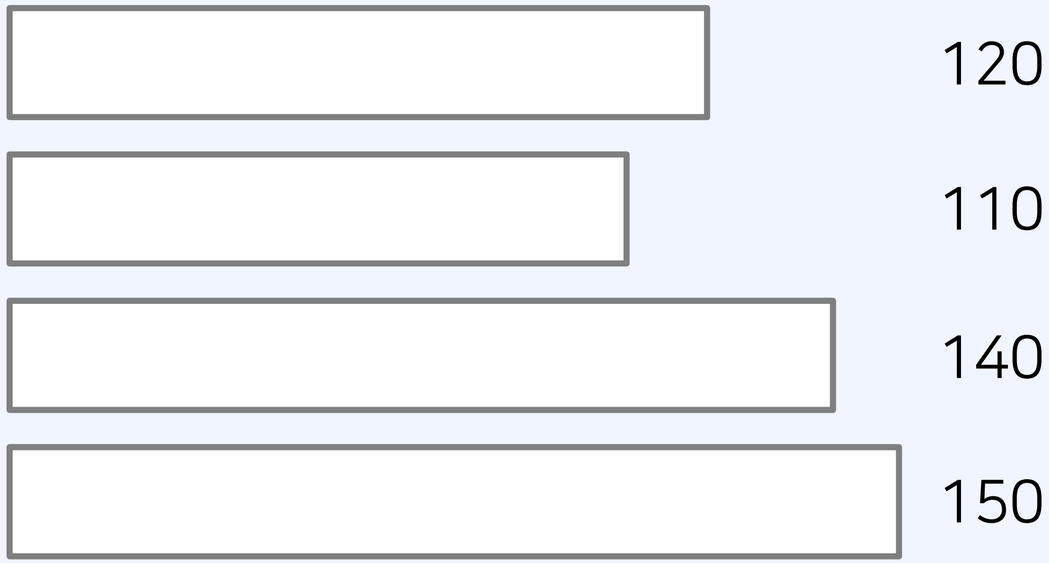
추천 풀이 시간: 30분

예산: https://www.acmicpc.net/problem/2512

문제 해결 아이디어

JavaScript
이진 탐색
파라메트릭
서치

- **문제 요구사항**: 적절한 **상한 금액**을 찾는 것이 문제의 목표다.
- 전체 국가 예산이 485이고, 4개의 지방 예산 요청이 120, 110, 140, 150이라고 하자.

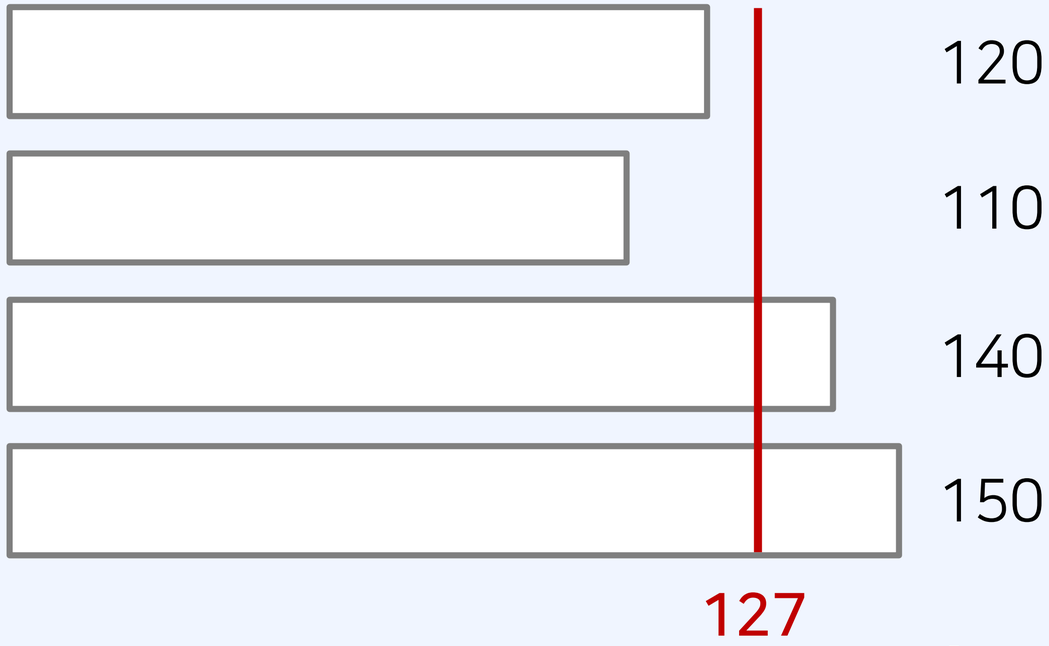


예산: https://www.acmicpc.net/problem/2512

문제 해결 아이디어

JavaScript
이진 탐색
파라메트릭
서치

- **문제 요구사항**: 적절한 **상한 금액**을 찾는 것이 문제의 목표다.
- 전체 국가 예산이 485이고, 4개의 지방 예산 요청이 120, 110, 140, 150이라고 하자.
- 상한 금액이 127인 경우, 배정 금액의 합이 120 + 110 + 127 + 127 = 484이다.



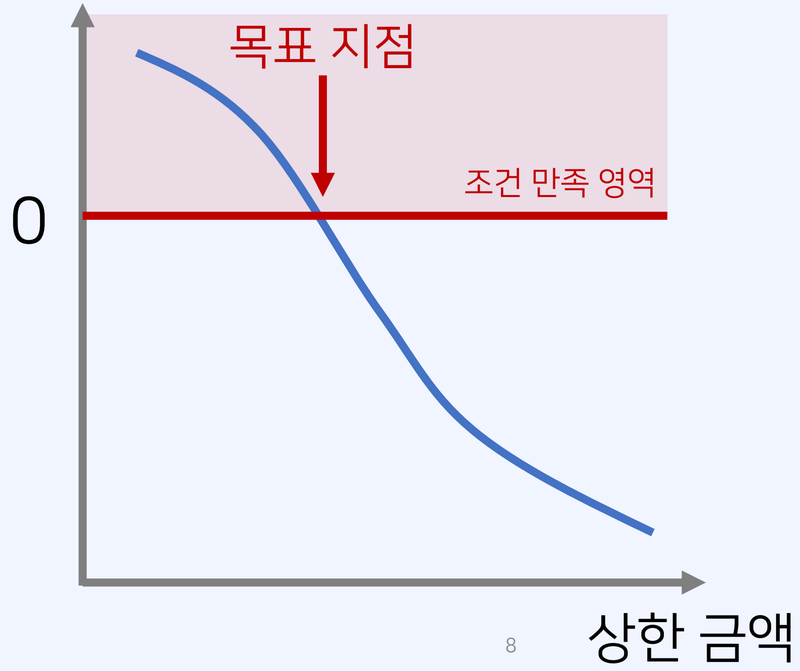
예산: https://www.acmicpc.net/problem/2512

문제 해결 아이디어

JavaScript
이진 탐색
파라메트릭
서치

- **문제 요구사항**: 적절한 **상한 금액**을 찾는 것이 문제의 목표다.

1. 배정된 총 예산이 조건을 만족한다면, 상한 금액을 증가(최대화가 목표)시킨다.
2. 배정된 총 예산이 조건을 만족하지 못한다면, 상한 금액을 감소시킨다.



예산: https://www.acmicpc.net/problem/2512

정답 코드 예시

JavaScript
이진 탐색
파라메트릭
서치

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

let n = Number(input[0].split(' ')[0]); // 지방의 수(N)
let arr = input[1].split(' ').map(Number); // 각 지방의 예산 요청
let m = Number(input[2]); // 총 예산(M)

let start = 1; // 이진 탐색을 위한 시작점(start)과 끝점(end) 설정
let end = arr.reduce((a, b) => Math.max(a, b));

let result = 0;
while (start <= end) { // 이진 탐색 수행(반복문)
  let mid = parseInt((start + end) / 2); // 현재의 중간점(상한액)
  let total = 0; // 배정된 예산의 총액 계산
  for (x of arr) { // 각 지방에서 요청한 예산을 하나씩 확인하며
    total += Math.min(mid, x); // 예산 배정
  }
  if (total <= m) { // 조건을 만족한다면, 상한액(최대화가 목표)을 증가시키기
    result = mid;
    start = mid + 1;
  }
  else { // 조건을 만족하지 못한다면, 상한액을 감소시키기
    end = mid - 1;
  }
}
console.log(result);
```