

Binary Search

스터디 2주차 - 최적해 도출

문종건

fragmentbombs@gmail.com

문제 선정 기준



1. 최적해 탐색

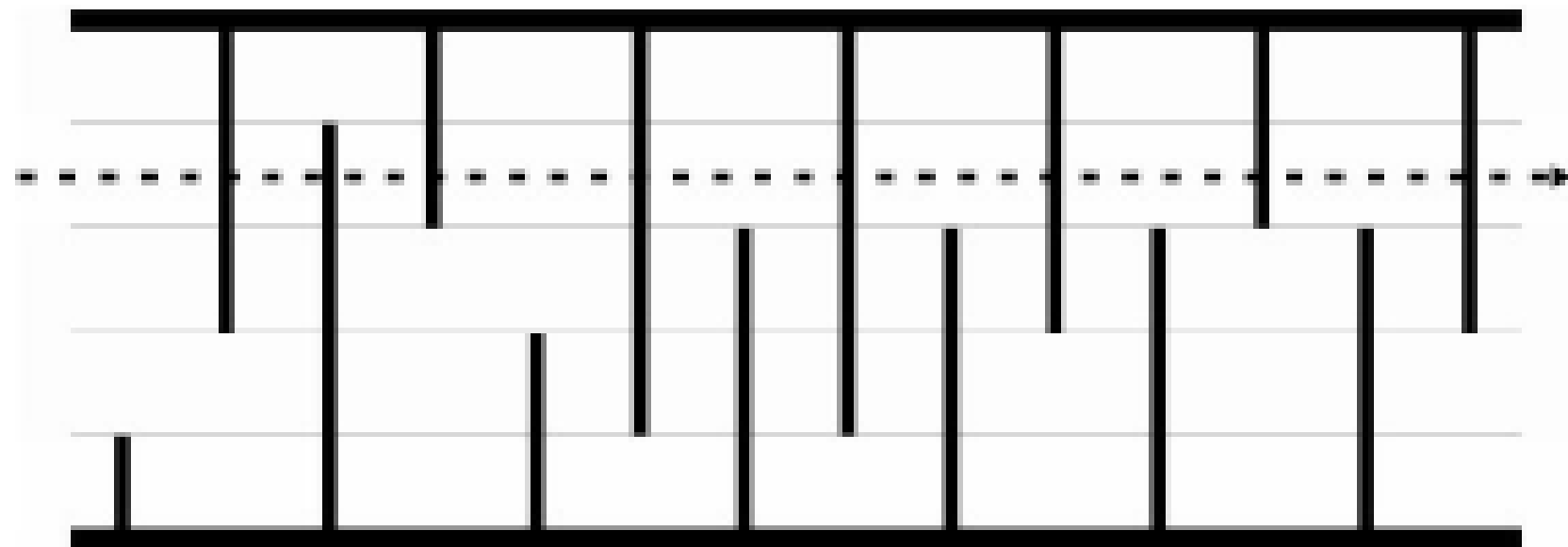
특정 값을 찾음으로써 풀 수 있는
공통 문제들로 구성



2. 풀이의 다양성

이분 탐색 베이스의 메인 풀이와,
다른 방식으로 생각해도 풀 수 있
는 문제들

3020(G5) 개똥벌레



문제

파괴해야 하는 장애물의 최소값과, 그러한 최소 구간이 몇개 있는지 구하시오.

조건

$2 \leq \text{장애물 개수} \leq 200,000$

$2 \leq \text{동굴 높이} \leq 500,000$

풀이

찾을 수 있는 다양한 방법을 생각해낼 수 있는 것이 중요!

종유석의 경우

1 | 2 | 0 | 1 | 0

1 | 3 | 3 | 4 | 4

1.Brute Force

1부터 H까지, 모든 구간에 대해 부숴야 할 장애물의 개수를 각각 체크

$O(HN)$ - 시간 초과

2.Binary Search

종유석과 석순을 다른 배열에 저장하고, 각각 높이에 따라 정렬 -> 이후 각 구간에 대해 부숴야 할 장애물의 개수를 이분 탐색으로 파악 가능

$O(H \log N)$ - 맞았습니다!

3.Prefix Sum

종유석과 석순을 위한 두개의 배열을 준비

입력 시 각각 높이를 index삼아 해당 칸에 그 높이의 장애물 개수를 저장

이후 종유석/석순 여부에 따라 배열 전체를 다른 순서로 순회, 배열의 이전 값을 현재 값에 더하여 누적합 배열 생성

->이후 각 구간에 대해 부숴야 할 장애물의 개수를 누적합 배열을 통해 $O(1)$ 에 파악 가능

$O(H)$ - 맞았습니다!

2467(G5) 용액



문제

두 값을 더했을때 가장 0에 가까운 값들을 구하시오.



조건

$2 \leq \text{배열 길이} \leq 100,000$

풀이

-99 | -2 | -1 | 4 | 98

1.Brute Force

모든 쌍을 전부 체크 - NC2가지
 $O(N^2)$ - 시간 초과

2.Binary Search

두 값중 하나를 미리 고정해두고, 해당 값에 대해서 더했을 때 가장 0에 가까워지는 값을 이분 탐색 -> x 와 -x?
모든 값에 대해 고정 후 이분탐색 과정을 반복
 $O(N \log N)$ - 맞았습니다!

3.Two Pointers

배열을 우선 정렬하고, 처음과 끝을 가리키는 "포인터"를 사용
 $l < r$ 인 동안, 두 값의 합이 현재까지 찾은 최적해보다 더 0에 가까운지를 체크하고, 최적해 업데이트
이후 0보다 작으면 $l++$, 크면 $r--$ 하여 합의 부호가 바뀔 수 있는 방향으로 진행
 $O(N)$ - 맞았습니다!

14002(G4) 가장 긴 증가하는 부분 수열 4(LIS)

문제

가장 긴 증가하는 부분 수열의 길
이와, 해당 수열을 구하시오

조건

$1 \leq \text{수열의 길이} \leq 1,000$

$1 \leq \text{수열의 원소} \leq 1,000$

풀이

생각하기에 따라 풀이가 복잡해질 수도 있다.

1. Dynamic Programming - $O(N^2)$

$dp[i]$ -> 배열 처음부터 i 번 원소까지 LIS 길이
배열 처음부터 각 원소 별로 해당 원소 전까지의 배열을 다시 전부 탐색, dp 값은 현재 원소보다 작은 값의 원소들의 dp 값중 최대치 + 1
최대 길이를 달성한 원소 번호를 미리 저장해두고, 최대 길이를 구하는게 끝나면 뒤에서부터 앞으로 되돌아가며 수열을 탐색. (이전 원소가 현재 원소보다 작고, dp 값이 정확히 1 작으면, 그 원소는 최대 증가 수열에 포함된다.)

2. DP + Binary Search - $O(N \log N)$

$dp[i]$ -> LIS의 최소 i 번째 원소 값(?)
 dp 배열 자체는 항상 증가하는 수열 형태로 유지
원본 배열을 순회하면서 dp 배열의 `lower_bound`(삽입 시 배열이 정렬된 상태를 유지할 수 있는 위치)를 찾고, 해당 위치의 dp 배열값을 업데이트 하거나 dp 배열을 확장.
 dp 배열의 길이가 즉 LIS의 길이가 된다.
수열 구성은? -> 이후 설명

17951(G4)

훔날리는 시험지 속에서 내
평점이 느껴진거야

제목 쓸데없이 기네

문제

K개의 그룹으로 수열을 나누었을 때,
각 그룹의 합의 최솟값의 최대값을
구하시오.

조건

$2 \leq \text{구간 수}$, 배열 길이 $\leq 100,000$
 $0 \leq \text{수열 값} \leq 20$

풀이

사실 맞았습니다! 를 받을 수 있는 풀이가 다양하진 않지만
이분 탐색(매개 변수 탐색)의 응용 기초 문제

한양대학교 프로그래밍 경시대회 기출이라 나름 문제 설명도 정감가고..

1.Brute Force

K개 구간으로 나눌 수 있는 모든 구간에 대해 탐색
 $O((N-1)C(K-1))$ - 시간 초과

2.Parametric Search

간단하게, 답이 되는 값 자체를 이분탐색으로 찾는 접근법
최적해 = 모든 구간 별 합의 최솟값이므로, 현재까지 찾은
값에 대해 배열이 그 값 이상의 합을 가진 K개 이상 구간으
로 나뉠 수 있는지 이분 탐색 과정에서 매번 확인.

불가능-> $r=mid$, 가능-> $l=mid$;

$O(N \log S)$ (S: 배열 전체 합) - 맞았습니다!

14003(P5)

가장 긴 증가하는 부분 수열
5

문제

가장 긴 증가하는 수열의 길이와, 해당 수열을 구하시오.

조건

$1 \leq \text{배열 길이} \leq 1,000,000$

$-1e9 \leq \text{수열 값} \leq 1e9$

풀이

전에 쓰였던 DP식의 정의를 재활용 할 수도 있다.

별도로 Segment Tree를 활용하여 LIS의 길이를 같은 시간 복잡도에 구할 수 있는 방법도 있지만, 수열 복원 과정은 같다.

1.DP + Binary Search

앞서 LIS 4에서 본 $O(N \log N)$ DP를 활용하여 길이는 구할 수 있다.

수열까지 구하려면... 이전에 사용했던 $O(N^2)$ DP배열의 정의도 재활용.(현재 원소를 마지막으로 하는 LIS의 길이)

두 DP배열을 모두 사용하되, DP 과정은 $O(N \log N)$ 에 맞추고, 이전에 사용했던 두번째 DP배열은 $O(N \log N)$ DP배열이 채워질때 함께 갱신(별도의 탐색 x)

결국 시간은 단축하되, 같은 DP배열을 얻었으므로 이전 문제와 같은 과정을 통해 LIS를 구할 수 있다!

2.Segment Tree?