

늘어난 정복 (오픈)

소스코드

최백준 choi@startlink.io



$$\text{inorder}[\text{position}[i]] = i$$

C++14

```

1 #include <stdio>
2 int inorder[100000];
3 int postorder[100000];
4 int position[100001];
5 void solve(int in_start, int in_end, int post_start, int post_end) {
6     if (in_start > in_end || post_start > post_end) return;
7     int root = postorder[post_end];
8     printf("%d ", root);
9     int p = position[root];
10
11     // inorder: in_start p in_end
12     // postorder: post_start post_end
13     // left: p - in_start
14     // right: in_end - p
15     int left = p - in_start;
16     solve(in_start, p-1, post_start, post_start + left - 1);
17     solve(p+1, in_end, post_start+left, post_end-1);
18 }
19 int main() {
20     int n;
21     scanf("%d", &n);
22     for (int i=0; i<n; i++) scanf("%d", &inorder[i]);
23     for (int i=0; i<n; i++) scanf("%d", &postorder[i]);
24     for (int i=0; i<n; i++) {
25         position[inorder[i]] = i;
26     }
27     solve(0, n-1, 0, n-1);
28     return 0;
29 }

```

인덱스
[배열값이
인덱스에서 몇번인지]

루트의 위치
(인덱스)

왼쪽 자식
오른쪽 자식

인덱스

인덱스

결과

메모리

시간

코드 길이

맞았습니다!!

6332 KB

44 ms

834 B

끝

코드 플러스

<https://code.plus>

- 슬라이드에 포함된 소스 코드를 보려면 "정보 수정 > 백준 온라인 저지 연동"을 통해 연동한 다음, "백준 온라인 저지"에 로그인해야 합니다.
- 강의 내용에 대한 질문은 코드 플러스의 "질문 게시판"에서 할 수 있습니다.
- 문제와 소스 코드는 슬라이드에 첨부된 링크를 통해서 볼 수 있으며, "백준 온라인 저지"에서 서비스됩니다.
- 슬라이드와 동영상 강의는 코드 플러스 사이트를 통해서만 볼 수 있으며, 동영상 강의의 녹화와 다운로드, 배포와 유통은 저작권법에 의해서 금지되어 있습니다.
- 다른 경로로 이 슬라이드나 동영상 강의를 본 경우에는 codeplus@startlink.io 로 이메일 보내주세요.
- 강의 내용, 동영상 강의, 슬라이드, 첨부되어 있는 소스 코드의 저작권은 스타트링크와 최백준에게 있습니다.