

11055번: 가장 큰 증가 부분 수열

11055번 제출 맞은 사람 슛코딩 풀이 풀이 작성 풀이 요청 재채점/수정

문제 추천 채점 현황 내 소스 강의▼ 질문 검색 질문 작성

가장 큰 증가 부분 수열 풀이

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	256 MB	5634	2480	2048	46.598%

문제

수열 A가 주어졌을 때, 그 수열의 증가 부분 수열 중에서 합이 가장 큰 것을 구하는 프로그램을 작성하시오.

예를 들어, 수열 $A = \{1, 100, 2, 50, 60, 3, 5, 6, 7, 8\}$ 인 경우에 합이 가장 큰 증가 부분 수열은 $A = \{1, 100, 2, 50, 60, 3, 5, 6, 7, 8\}$ 이고, 합은 113이다.

처음에 생각한 풀이는

n 길이의 수열중 n 이 선택 되거나 선택하지 않거나를 선택하게 하는 리커시브 함수를 생각 하였다.

$f(n) = \max(a(n) + s(n) , f(n-1))$

하지만 $s(n) =$ 이라는 서브 함수를 호출하게 된다. 이 서브 함수는

$s(n) = a(n)$ 부터 시작할때 가장 합이 큰 부분 수열.

$S(1)$ 부터 $S(n)$ 까지 모두 작성한다면, 굳이 $f(n)$ 이라는 함수가 필요 없어지게 된다. $S(n)$ 을 돌린 후 제일 큰 값 출력.

이 문제를 풀면서 실수 한 것은 문제를 sub-problem으로 완벽하게 나누었다고 착각한 것이다. 나누기야 나누었지만, 효율적인 분리가 아니었다.

```
#include <stdio.h>
```

```
int main(void){
    int N;
    int Dp[1010] = {};
    int A[1010] = {};
    int max = 0;
    scanf("%d", &N);

    for (int i = 1; i <= N; i++)
        scanf("%d", &A[i]);

    for (int i = 1; i <= N; i++)
    {
        int maxDp = 0;
        for (int j = 0; j < i; j++){
            if (A[i]>A[j]){
                if (maxDp < Dp[j])
                    maxDp = Dp[j];
            }
        }
        Dp[i] = maxDp + A[i];
        if (max < Dp[i])
            max = Dp[i];
    }

    printf("%d\n", max);
```

```
}
```

lower Bound 로 풀 수 있다.

입력

첫째 줄에 수열 A의 크기 N ($1 \leq N \leq 1,000$)이 주어진다.

둘째 줄에는 수열 A를 이루고 있는 A_i 가 주어진다. ($1 \leq A_i \leq 1,000$)

출력

첫째 줄에 수열 A의 합이 가장 큰 증가 부분 수열의 합을 출력한다.

예제 입력

```
10
1 100 2 50 60 3 5 6 7 8
```

예제 출력

```
113
```

힌트

출처

- 문제를 만든 사람: baekjoon
- 데이터를 추가한 사람: gomyk12

알고리즘 분류

[보기](#)