

<https://www.acmicpc.net/problem/1932>

1932번 숫자삼각형 문제풀이

# DP = Dynamic Programming

Q.다이나믹 프로그래밍(동적 계획법)이 뭘까?

A.이전에 저장한 데이터(이전에 구한 결과)를 바탕으로 또 다시 결과를 구하는  
효율적인 알고리즘!

Q.그럼 뭘 어디다가 저장해야 되는데?

A.dp테이블을 별도로 만들어서 dp테이블에 상태값을 정의하고 상태값을 저장  
해야된다!

```

6  int main() {
7      int n;
8      //arr[i][j] 는 i층 j번째 수를 뜻한다.
9      int arr[510][510];
10     scanf("%d", &n);
11     for (int i = 0; i < n; ++i) {
12         for (int j = 0; j <= i; ++j) {
13             scanf("%d", &arr[i][j]);
14         }
15     }
16     //dp 테이블. dp[i][j]는 i층 j번째 수를 더했을 때의 최대값을 뜻한다.
17     int dp[510][510];
18     //정답 변수
19     int maxi = 0;
20     //처음에 필요한 데이터(때에 따라서 몇 개가 될지는 모름)는 미리 입력해줘야한다.
21     dp[0][0] = arr[0][0];
22     //알고리즘
23     for (int i = 1; i < n; ++i) {
24         for (int j = 0; j <= i; ++j) {
25             if (j == 0) {
26                 //dp[i][0]는 맨 위층부터 i층까지의 제일 왼쪽에 있는 수들의 합이다.
27                 dp[i][0] = dp[i - 1][0] + arr[i][0];
28             } else {
29                 dp[i][j] = max(dp[i - 1][j], dp[i - 1][j - 1]) + arr[i][j];
30             }
31             if (i == n - 1) {
32                 //맨 아래층까지 내려오면 maxi에 정답값을 대입해준다.
33                 maxi = max(maxi, dp[i][j]);
34             }
35         }
36     }
37     //정답 출력
38     printf("%d", maxi);
39 }

```

# 알고리즘 이론적 분석

시간 복잡도 :  $O(n)$ ( $n$ 개의 수가 입력되었을 경우.)

ex) 5층  $\Rightarrow 1+2+3+4+5 = 15$

$n = 15$