사칙연산

# Index	1843
CreatedAt	@September 28, 2022
Person	A Ally Hyeseong Kim
Status	Done
Tags	Dynamic Programming Python
UpdatedAt	@September 28, 2022

References

https://school.programmers.co.kr/learn/courses/30/lessons/1843

1. Dynamic Programming

```
def solution(arr):
    arr[0] = int(arr[0])
    for i in range(1, len(arr) // 2 + 1):
        if arr[i] == "+":
            arr = arr[:i] + [int(arr[i + 1])] + arr[i + 2:]
        elif arr[i] == "-":
            arr = arr[:i] + [-int(arr[i + 1])] + arr[i + 2:]

    max_dp = [[-float("inf") for _ in range(len(arr))] for _ in range(len(arr))]
    min_dp = [[float("inf") for _ in range(len(arr))] for _ in range(len(arr))]

    for length in range(len(arr)):
        for i in range(len(arr) - length):
            j = i + length
            if length == 0:
                max_dp[i][j] = abs(arr[i])
                min_dp[i][j] = abs(arr[i])

            for k in range(i + 1, j + 1):
                if arr[k] > 0:
                    max_dp[i][j] = max(max_dp[i][j], max_dp[i][k - 1] + max_dp[k][j])
```

```
                    min_dp[i][j] = min(min_dp[i][j], min_dp[i][k - 1] + min_dp[k][j])
                else:
                    max_dp[i][j] = max(max_dp[i][j], max_dp[i][k - 1] - min_dp[k][j])
                    min_dp[i][j] = min(min_dp[i][j], min_dp[i][k - 1] - max_dp[k][j])

    return max_dp[0][len(arr) - 1]
```