

{% note info %} **摘要** Title: CF1155 D. Beautiful Array Tag: dp、最大子段和 Memory Limit: 64 MB Time Limit: 1000 ms {% endnote %}

Powered by:NEFU AB-IN

[Link](#)

@TOC

## D. Beautiful Array

---

- 题意

给出一个长度为 $n$ 的数列和数字 $x$ ，经过最多一次操作将数列中的一个子段都乘 $x$ ，使该数列的子段和最大

- 思路

一般看到子段和，就要想到**最大子段和**， $dp[i]$ 表示以 $i$ 结尾的最大子段和，每次取 $\max$ 即可 **最大子段和**

```
'''
Author: NEFU AB-IN
Date: 2021-11-05 19:16:12
FilePath: \ACM\test.py
LastEditTime: 2022-03-05 17:25:05
'''

N = int(100)
INF = int(2e9)
dp, a = [0] * N, [0] * N

n = int(input())
a[1:] = map(int, input().split())

maxn = 0
for i in range(1, n + 1):
    dp[i] = max(0, dp[i - 1] + a[i]) #每次判断是重新开始，还是继承上一个，保证
    dp[i-1] > 0
    maxn = max(maxn, dp[i])

print(maxn)
```

---

很明显此题也是**最大子段和**的变式 将数列分为**三段**考虑

- 未修改
- 乘 $x$
- 未修改

设  $dp[i][j]$  为以  $i$  结尾的第  $j+1$  段的最大子段和

那么

- 第一段  $dp[i][0] = \max(0, dp[i-1][0] + a[i])$
- 第二段  $dp[i][1] = \max(a[i] * x, dp[i-1][0] + a[i] * x, dp[i-1][1] + a[i] * x)$ 
  - ps: 是要考虑前一段的, 即前面的全要考虑, 因为不知道上一个点是什么段的
- 第三段  $dp[i][2] = \max(a[i], dp[i-1][0] + a[i], dp[i-1][1] + a[i], dp[i-1][2] + a[i])$

可以发现递推式中有重复的, 所以可以改成

- 第一段  $dp[i][0] = \max(0, dp[i-1][0] + a[i])$
- 第二段  $dp[i][1] = \max(dp[i][0], dp[i-1][1] + a[i] * x)$
- 第三段  $dp[i][2] = \max(dp[i][1], dp[i-1][2] + a[i])$

## • 代码

```
...
Author: NEFU AB-IN
Date: 2022-03-05 18:58:47
FilePath: \ACM\Codeforces\1155\d.py
LastEditTime: 2022-03-05 22:56:34
...

N = int(3e5 + 10000)
dp = [[0] * 3 for _ in range(N)]
a = [0] * N
INF = int(2e9)

n, x = map(int, input().split())
a[1:] = list(map(int, input().split()))
maxn = 0

for i in range(1, n + 1):
    dp[i][0] = max(0, dp[i-1][0] + a[i])
    dp[i][1] = max(dp[i][0], dp[i-1][1] + a[i] * x)
    dp[i][2] = max(dp[i][1], dp[i-1][2] + a[i])
    maxn = max(maxn, dp[i][2])

print(maxn)
```