

monotone stack

在经典的一道单调栈例题中: 给出一个数组, 求出每一个数组的左边的和它相比最小的数, 如果用到朴素算法, 很容易想到用双层循环: 枚举每一个数, 接着从数组头枚举到这个数, 找到左边的和他相比最小的数, 时间复杂度是 $O(n \log n)$.

当我们模拟这一过程的时候, 我们会发现, 当我们要求第三个元素的目标值时, 我们从头开始枚举, 但是第一个元素我们在考虑第二个元素的目标值时就已经处理过了: 我们做了重复操作.

```
#include <iostream>

int main () {
    std::cout << "HELLO WORLD\n";
}
```