

删除一次得到子数最大和。可以选择删也可以不删。

$arr = [1, -2, 0, 3]$

Step 1: 选出一个合适子集 \rightarrow 不删除情况下的最大子数和。(需连续)

Step 2: 删除最小的一个数。

解:

① 定义 $f(i)$: 不删除情况下的最大子集, 以 $arr[i]$ 结尾
 \rightarrow 代表数组中的 index.

$g(i)$: 删除情况下, 同样以 $arr[i]$ 结尾。

② 初始化: $dp[i][0]$ 不删
 $dp[i][1]$ 删 (删了 i , 不删 i)

③ 1° 不删: 比较 $dp[i-1][0] + arr[i]$, $arr[i]$.

前面的数如果 > 0 可以加上
 < 0 直接清空改 $arr[i]$.

2° 删: 1° 删除 $arr[i]$ 位置: $dp[i+1][1] = dp[i][0]$

删除其他位置: $dp[i+1][1] = dp[i][1] + arr[i]$.

比如 $[-1, 2, -3]$ 删了 -3 .

3° 比较删和不删。

1° $[1, -2, 0, 3]$.

$i = -1$ ($-\infty, -\infty$)

$i = 0$ ($1, -\infty$) $ans = 1$.

$i = 1$ ($-1, 1$) $\rightarrow [1, -2]$ 把 -2 删了。

$\Gamma = 2 \quad (0, 41) \rightarrow \text{不在 } [1, -2, 0] \text{ 中.}$
 $\Gamma = 3 \quad (3, 4) \rightarrow \text{不在 } [1, -2, 0, 3] \text{ 中.}$