

# PA1-4 Jump

---

## 使用数据结构

单调队列

## 算法构思

此题通过题意便可看出是一个动态规划问题，倒序分析，易知问题的核心是求一个区段内的最小值。

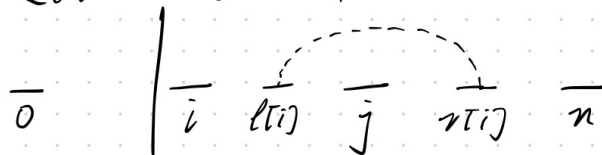
# Jump. 单调队列优化.

基操: 遍历  $i$  in  $[0, n)$ .

遍历  $j$  in  $[l[i], r[i]]$ .

更新  $T[j]$  为  $\min\{T[j], T[l[i]] + t[l[i]]\}$ .

DP. 倒序遍历  $i$  in  $[0, n)$ .



for ( $i = n-1; i \geq 1; i--$ )

只找  $[i, n)$  的最短时.

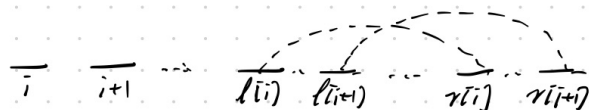
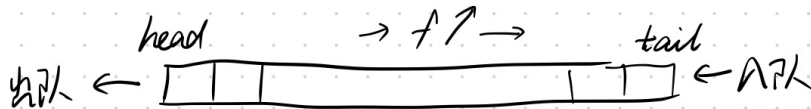
(已维护好  $[i+1, n)$  的最短时)

对 find  $j$  in  $[l[i], r[i]]$ , s.t.  $f[j]$  最小.

更新  $f[i] = t[i] + f[j]$ .

优化. 注意到  $l_i, r_i$  单↑.

若  $j \geq l_i \geq l_{i-1} \geq \dots l_1$ .



关键: 若  $f[j+k] \geq f[j]$

$j+k$  直接舍弃

( $l_i \leq j \leq j+k \leq r_i$ )

$[l[i], l[i+1])$  入队 (大的先入)

每入一个: 检查出队

若  $f[tail] \geq f[j]$

$tail$  出队 (队尾出)

插入 $h$ : 0'空

②  $f[tail] < f[i]$   
实现单调性

出队: 若  $head > r[i]$  出.

直接取  $f(head)$  即可

由单调队列只入队一次出队一次的特点, 此题复杂度降至线性复杂度。

维护队列的单调性是关键, 且由此, 单调队列可用于任何需要取极值, 且FIFO的场景。

## 遇到的问题及debug过程

动态规划的优化是这道题考察的地方, 单调队列是个之前没有接触过的概念, 因此在理解单调队列的原理上花了比较多的时间。

实现过程中, 有一个小问题花了比较多的时间调试, 就是define 无穷大INF时, 沿用了int时使用的0x3f3f3f3f, 但这道题的数据应使用long long, 所以会造成INF不够大, 少数点WA的错误, 在意识到这个问题后放弃使用INF, 顺利通过。

## 参考资料

董博文小教员指出了构造单调队列的思路。

PPT中的队列实现。

## 复杂度

时间:  $O(n)$

空间:  $O(n)$