

## V题题解报告

很有意思的一个题目，让我想到了数据结构专题的一个糖罐的题目；

做题体验挺好地，解题思维上是很舒服地递进的；

首先，题目有个信息讲的不是很清楚：每张票不一定非要坐满  $r$  个站，可以中途下站。

题目让我们求最低VIP票价；

试想一下，假如第  $i$  张票的价格小于第  $i - 1$  的票的，那么我们就可以不用考虑第  $i - 1$  这张票了。

因为我们有票价更低，功能更强大的票可以代替它；

所以这里就要预处理下，每次读入价格更小的票就把前面比这个票价格高的去除掉；

预处理后，二分枚举票，

对于每个枚举到的票，用  $dp[i]$  表示到第  $i$  个站的最少耗时；

转移方程：

$$dp[i] = dp[i-j] + cost[j]$$

这里要用到单调队列优化，不然会超时

转移方程变为：

$$dp[i] = stack[head] + Time\_cost[i];$$

维护单调队列：

```
while(head <= tail && stack[tail] > dp[i]) tail--;  
stack[++tail] = dp[i];  
pos[tail] = i;  
while(pos[head] <= i - move && head < tail) head++;
```

之所以用到单调队列，是因为，对于第  $i$  个站的最少耗时

我们只要找到前某些站的最小值就行了

于是就用这个单调队列优化，实现  $O(1)$  的查询

再提下枚举车票时二分而不是依此枚举：

如果很不幸票在偏后的位置，预处理并没有处理掉太多车票，枚举太多容易超时。

二分的话，我们枚举的那个如果可以那就向前找（后面的票一定满足时间要求，但是我们要找票最小的）

噢噢这里忘了说了，最后一定是只选一张票。如果选了两张票的话后面一张票完全可以代替前一张，选两张就很没必要。

时间复杂度 $O(n \log n)$ :

二分枚举票,  $\log n$ , 单次枚举遍历所有站点,  $n$