

# queue 解题报告

江苏省常州高级中学 徐毅

October 7, 2013

## 1 题目大意

有  $n$  个主题的请求，第  $i$  个主题有编号  $a_i$ ，请求数  $b_i$ ，第一个请求到达时间  $c_i$ ，每个请求的处理时间  $d_i$  和相邻请求到达的间隔时间  $e_i$ 。有  $p$  个人来处理这些请求，第  $i$  个人有编号  $f_i$  和  $g_i$  个可以处理的主题  $h_{i,j}$ ，主题列表按优先级从高到低的顺序用主题编号给出。一个人空闲时会从自己的主题列表里按优先级从高到低选择一个已到达的请求开始处理，上次开始处理时间早的人优先选择，上次开始处理时间相同则编号小的人优先选择。求处理完所有请求的总时间  $m$ 。

## 2 数据规模和约定

30%  $1 \leq m \leq 10000$ 。

100%  $1 \leq g_i \leq n \leq 20$ ,  $1 \leq p \leq 5$ ,  $1 \leq b_i \leq 1000$ ,  $1 \leq d_i, e_i, m \leq 2^{31} - 1$ ,  
 $0 \leq a_i, c_i, f_i, h_{i,j} \leq 2^{31} - 1$ 。

## 3 30% 的算法

为方便叙述，我们令  $r = \sum_{i=1}^n b_i$ ,  $q = \sum_{i=1}^p g_i$ 。

首先，对于编号到主题的映射，可以用哈希表或者 C++ 的 map 来处理。

由于  $m$  较小，我们可以考虑逐时刻模拟选择过程。

对于第  $i$  个人，我们需要记录其当前开始处理时间  $last_i$  和下次空闲时间  $free_i$ 。

对于时刻  $t$ ，我们把满足  $free_i \leq t$  的空闲的人拿出来，按  $last_i$  为第一关键字  $f_i$  为第二关键字从小到大进行排序以得到空闲的人的选择顺序。

第  $i$  个人选择时，按优先级从高到低访问其主题列表，找到一个已到达的请求，修改  $last_i$  和  $free_i$ ，同时修改对应主题已处理的请求个数。

当所有主题的请求都处理完毕时， $\max_{i=1}^p free_i$  即为答案。

时间复杂度为  $O(m(p \log p + q) + r + q \log r)$ ，空间复杂度为  $O(n + q)$ 。

## 4 100% 的算法

选择的过程可完全参照 30% 的算法。

由于  $m$  几乎无限制，我们必须使算法的时间复杂度和  $m$  无关。

我们发现，只有某个请求到达或者某个人处理完某个请求开始空闲的时刻是可能进行选择的，而这样的有效时刻不超过  $2r$  个。

为了找出这样的有效时刻，我们可以用一个优先队列来进行实时维护，队头即为下一个有效时刻。

初始时，每个主题的第一个请求到达时间被加入优先队列。

当一个人选择后，对应主题的下一个请求到达时间（假如存在的话）和这个人的下次空闲时间被加入优先队列。

当队列为空时处理完毕，最后一个有效时刻即为答案。

时间复杂度为  $O(r(p \log p + q + \log r) + q \log r)$ ，空间复杂度为  $O(r + q)$ 。