分块方法的应用

王子昱

1. 引言

什么是分块

定义？

本文讨论的问题：

- 将维护的对象分为若干块，在每块内维护某些信息。
- 将要处理的问题分为规模小于和大于某个阈值的情况，并分别处理两种情况。

一些经典的分块问题

1. 离散模对数.
2. RMQ的$O(NloglogN) - O(loglogN)$做法.
3. SPOJ UNTITLE1.

2. Problem 3：糖果公园

题意

给定一棵点带权的树。Q组操作，每次可以修改一个点的权值，或是询问连接点s_i, t_i 的路径的权值。

其中路径的权定义如下：给定$V[]$，$W[]$，设路径上第i种权值W_i的出现次数为N_w，则该路径的权值为

$$\sum_{i}\sum_{j=1}^{N_w} W_i V_j$$

题意

给定一棵点带权的树。Q组操作，每次可以修改一个点的权值，或是询问连接点s_i, t_i 的路径的权值。

其中路径的权定义如下：给定$V[]$，$W[]$，设路径上第i种权值W_i的出现次数为N_w，则该路径的权值为

$$\sum_{i}\sum_{j=1}^{N_w} W_i V_j$$

点数N，操作数Q不超过10^5。

题意

数据分为三类：

1. 树是一条链，无修改。
2. 树不是链，无修改。
3. 树不是链，带修改。

在线算法。

链的情况

1	2	3	4	5	6	7	8	9	10	11	12
*			*			*			*		

将链分为$\sqrt{N}$块，每块的第一个元素称为关键点。预处理出每对关键点之间的答案。

考虑一次询问(l,r)。找到在区间内离端点最近的两个关键点L, R，用$[l,L)$和$(R,r]$中的数更新答案。

链的情况

1	2	3	4	5	6	7	8	9	10	11	12
*			*			*			*		

涉及到的问题：查询一个权值在区间中的出现次数。

链的情况

1	2	3	4	5	6	7	8	9	10	11	12
*			*			*			*		

涉及到的问题：查询一个权值在区间中的出现次数。

对前i个数，维护每种权值的出现次数。由于第i个数的信息与第$i-1$个数的信息只有一个位置不同，可以使用持久化数组维护。

链的情况

1	2	3	4	5	6	7	8	9	10	11	12
*			*			*			*		

涉及到的问题：查询一个权值在区间中的出现次数。

对前i个数，维护每种权值的出现次数。由于第i个数的信息与第$i-1$个数的信息只有一个位置不同，可以使用持久化数组维护。

使用块状链表作为持久化数组的实现，可以得到$O(\sqrt{N})$的做法。

实现细节参见论文。

树的情况

对树上无修改的情况，我们可以把链上的算法直接推广到树上。

树的情况

对树上无修改的情况，我们可以把链上的算法直接推广到树上。

- 关键点的选取：按高度将树分块，取每块中深度最小的点
- 区间中权值的出现次数⇒路径上权值的出现次数：维护从根到每个点的持久化数组

树的情况

树上有修改的情况

一种做法：将树分为$n^{1/3}$块，维护每对关键点之间，每种权值的出现次数，每次修改时直接修改每对关键点之间的函数式数组

树的情况

树上有修改的情况

一种做法：将树分为$n^{1/3}$块，维护每对关键点之间，每种权值的出现次数，每次修改时直接修改每对关键点之间的函数式数组

- 空间复杂度至少为$O(n^{5/3})$，难以承受

另一种做法

树的情况：带修改

处理一次修改操作对询问的影响。

树的情况：带修改

处理一次修改操作对询问的影响。

```
Path : a   b   b   a   c   a   b
           ^
           c
new-answer = answer - freq[b] * W[b] + (freq[c] + 1) * W[c]
```

修改答案和临时数组$freq$。O(1)。

树的情况：带修改

一种做法：忽略所有修改操作，计算所有询问的答案，之后对每个询问，用在它之前的修改修正答案。

- 时间代价难以接受
- 没有必要这种方法处理所有修改

树的情况：带修改

一种做法：忽略所有修改操作，计算所有询问的答案，之后对每个询问，用在它之前的修改修正答案。

- 时间代价难以接受
- 没有必要这种方法处理所有修改

将操作分为若干块。处理每块的询问之前，首先把之前的修改应用在树上，并重建整个数据结构。

处理询问时，只用和它在同一块中的修改操作修正答案。

树的情况：时间复杂度

将操作分为A块，每次重建时将树分为B块时，复杂度为$O(A(NB+W_{parr})+\frac{Q^2}{A})$。

- W_{parr}为建立整个树的持久化数组的时间

取$A=Q^{1/3},B=N^{1/3}$，得到最优复杂度。

树的情况：持久化数组

如果继续用块状表实现持久化数组，每次重建的复杂度$W_{parr} = O(N^{3/2})$，难以接受

树的情况：持久化数组

如果继续用块状表实现持久化数组，每次重建的复杂度$W_{parr} = O(N^{3/2})$，难以接受

对块状表再次分块

- 将待维护的数组分成$N^{1/3}$块，每块再分为$N^{1/3}$块
- 每次重建的代价为$O(N^{1/3})$

树的情况：持久化数组

如图所示。

$W_{parr} = O(N^{4/3})$，算法的时间复杂度为$O(Q^{1/3}N^{4/3} + Q^{5/3})$，空间复杂度为$O(N^{4/3})$。

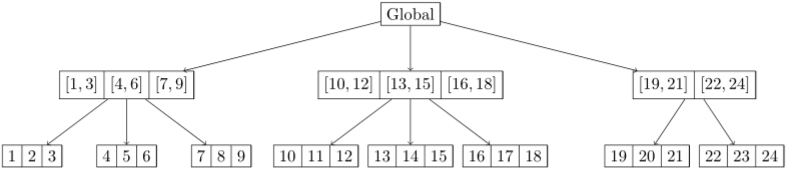

Figure: 块状表

3. Problem 4

题意

给定N个字符串集合，初始时每个集合中仅有一个字符串
操作：

1. MERGE A B 删除集合A、B，插入集合$A \cup B$。
2. QUERY S s 查询集合S中的串在字符串s中出现的次数。

$Slen < 10^5$。

模式长度较小的情况

维护$maxlen$个Hash表，表示某一固定长度的模式串的Hash值

询问时枚举模式串长度和开始位置

- $O(qlen \times maxlen)$

模式长度较大的情况

长度超过$\sqrt{Slen}$的串至多有$\sqrt{Slen}$个

对每个这样的串，暴力与询问串进行匹配

- 使用KMP
- 预处理这样的串的$next$数组之后，匹配的复杂度为$O(qlen)$
- 询问复杂度$O(\sqrt{Slen} \times qlen)$.

得到解法

结合上述两种做法

对于每个集合，对长度小于$\sqrt{Slen}$的串维护Hash表，对长度大于$\sqrt{Slen}$的串记录其标号

预处理每个长度大于$\sqrt{Slen}$的串的$next$数组

询问复杂度$O(\sqrt{Slen} \times qlen)$

合并时对每个Hash表以及标号数组进行启发式合并

- 总代价$O(NlogN)$

问题得到解决

4. Problem 6：Fairy

题意

给定一个图，询问存在多少边，使得删掉这条边之后原图为二分图

$N \leq 100000$，$M \leq 100000$

题解

判断一个图是否是二分图？

- 并查集

将边表分为$\sqrt{M}$块，对每一块，将不在这一块的所有边加入并查集，更新信息

对块内的每条边，将块内除它之外的边加入并查集，判断它是否为答案，再撤销加入块内边的操作

题解

判断一个图是否是二分图？

- 并查集

将边表分为$\sqrt{M}$块，对每一块，将不在这一块的所有边加入并查集，更新信息

对块内的每条边，将块内除它之外的边加入并查集，判断它是否为答案，再撤销加入块内边的操作

$O(N^{3/2}\alpha(N))$?

题解

基于平摊分析的复杂度，在这里不再适用

$O(N^{3/2}logN)$?

开始块内操作之前对所有点进行路径压缩

- 相当于缩点
- 前后两个过程独立
- $O(N^{3/2}\alpha(N))$

5. 小结

小结

$$(A + \frac{N}{A})_{min} = O(\sqrt{A})$$

- 均衡代价的思想

小结

$$(A+\frac{N}{A})_{min}=O(\sqrt{A})$$

- 均衡代价的思想

优势：通用性、高效率

- 高性价比的问题解决方案

谢谢大家

欢迎提问。