

《树与》命题报告

重庆市巴蜀中学 常瑞年

目录

1	题面	2
1.1	题目描述	2
1.2	限制与约定	2
2	题解	2
2.1	算法一	2
2.2	算法二	3
2.3	算法三	3
2.4	算法四	3
2.5	算法五	4
2.6	算法六	4
2.7	算法七	4
2.8	一些实现技巧	5
3	总结	5
4	验题	6
5	参考资料	6

1 题面

1.1 题目描述

给定一棵树,每次询问一条链上选出 m 个数与起来的最大值,强制在线

1.2 限制与约定

子任务	分值	$n \leq$	$m \leq$	$Q \leq$	$V \leq$	特殊性质
1	5	10^3	3	1		树是一条链
2	5			10^5		
3	10	10^5	2	1	65535	
4	5	10^6		10^4		
5	10	10^5				
6	15					
7	15	10^6		10^5		
8	35					

其中 V 表示权值范围,如表格中无特殊说明(即空白),则有 $n \leq 10^6$, m 在 $[2, 10]$ 之间随机生成, $Q \leq 10^5$, $0 \leq v_i \leq 2^{62} - 1$,树为普通的树

树按照某种随机方式生成的意思是,对于第 i 个点, $2 \leq i \leq n$,有 i 向 $[1, i - 1]$ 中均匀随机的一个数连边

2 题解

2.1 算法一

如果我们没有任何的思路,考虑从题目条件的定义入手写一个最简单的暴力,即使用搜索暴力枚举选取的集合,取最大值作为答案

这样做的时间复杂度为 $O(Qn^m)$,可以通过子任务1,获得5分

2.2 算法二

当 q 非常大的时候,注意到不同的询问之间其实有关系,考虑一个子集 $\{a_i\}$,若询问区间 $[l, r]$ 包含 $[\min\{a_i\}, \max\{a_i\}]$,则这个子集的值对询问有取 $\max$ 的贡献

那么我们枚举出了一个子集之后,可以使用类似二维前缀和的转移求出所有区间的答案,从而 $O(1)$ 回答询问

时间复杂度 $O(n^m + Q)$,可以通过子任务1, 2, 获得10分

2.3 算法三

考虑子任务3,根据一些熟知的技巧¹,我们知道可以使用折半的思想

即我们可以将每个数的二进制位分为前半部分和后半部分,并使用扫描线离线询问,枚举右端点,考虑插入一个数和询问一个数,使用一个二维大小不超过分成的两半二进制位值域的二维数组 $f[][]$ 维护

当插入一个数 $v = \overline{(ab)}_2$ 时(其中 a, b 分别表示前后半段),我们访问 $f[a][y]$,并将 $f[a][y]$ 对 $y \& b$ 取 $\max$,表示后面一个数的后半段为 y 时,前面一个数的前半段为 a 时的最大值

当查询 $v' = \overline{(a'b')}_2$ 时,我们访问 $f[x][b']$,并将答案对 $x \& a' * B + f[x][b']$ 取 $\max$,其中 B 表示前半段的最低位

这样,时间复杂度 $O((n + Q)\sqrt{V})$,可以通过子任务3,结合算法二可以获得20分

2.4 算法四

上述算法其实跟正解没有太大的关系,通常我们考虑二进制运算时会引入一种数据结构 $Trie$ 来辅助我们优化过程,而对于题目中的最大值我们往往按位处理

考虑对这个序列的元素建出一棵 $trie$ 树,并从高位到低位考虑答案,设当前答案在 $trie$ 的节点 x 上,若其右子树有着超过 m 个节点,则答案显然这一位可以取1, x 进入右子树,反之,答案这一位取不了1,我们可以忽略这一位,从而将右子树的节点去掉这一位重新插入到左子树中,并递归左子树获取答案

这样做的时间复杂度是 $O(n \log V + m \log^2 V)$,可以通过子任务3, 4, 结合算法二可以获得25分

¹参考<https://uoj.ac/problem/549>,在某年初赛试卷上的完善程序也有出现同样的思想

2.5 算法五

考虑把算法四的思路推广到 q 大的时候,类似区间第 k 大的思路, $trie$ 上维护的 $size$ 是可差分的信息,因此我们可以使用可持久化 $trie$,以对一个前缀建立一棵完整的 $trie$

对于一个询问 $([l, r], m)$ (此处把点对看做链上的区间),我们找到 r 对应的 $trie$ 的根和 $l - 1$ 对应的 $trie$ 的根,并将 ans 对应的节点在这两棵 $trie$ 上同时游走

考虑我们的算法,有的时候需要重新插入,有时需要询问当前子树大小,于是我们新开一棵临时 $trie$,表示重新插入的那些节点,并维护其 $size$,当前的子树大小可以通过一些加减简单算出

如果是树的情况,我们的算法仍能奏效,只需把链上差分改为树上差分,把链上的可持久化 $trie$ 弄到树上

这样做的时间复杂度是 $O(n \log V + Qm \log^2 V)$,能够通过子任务1, 2, 3, 4, 5, 6,从而获得50分

2.6 算法六

考虑随机树的情形,熟知,这种情况下 x, y 间路径的长度期望不会超过 $O(\log n)$,考虑降低 $qm \log^2 V$ 部分的复杂度,即给定数不多的情况下求答案

注意到,对于我们算法的过程,其实就是每次找到第 m 大的点,然后考察其当前位是否为1,如果是,则删去这一位不为1的点,否则,将其这一位抹去重新插入到 $trie$ 中

于是我们只需要这些数间的相对大小,每次取出最开头的 m 个数进行比较,于是我们利用一个 set 即可完成所有操作

对于子任务7,时间复杂度 $O(n + Qm \log V \log \log n)$,可以通过,结合前述算法,可以获得65分

2.7 算法七

通过模拟我们的算法,我们惊奇地发现这个问题有一个良好的性质,我们只会用到前 $(m - 1) \log V$ 大的数!

考虑证明,即通过 $trie$,我们维护一个备选集合,若向左子树递归,则加入右子树的不超过 $m - 1$ 个点,若向右,则备选集合不变,这样递归 $\log V$ 层后,备选集合的大小至多是 $(m - 1) \log V$

这样,如果我们能够快速找到链上前 K 大的值,我们就能将整个问题转为子任务6

这是一个经典的问题,即链上前 K 大,我们可以通过类似于NOI2010《超级钢琴》或者十二省联考《异或粽子》的思路,每次找到当前链上的最大值,并将当前链分裂为两条链,分别丢进堆中

为了快速实现分裂链和求得某一条链上的最大值,我们考虑一些经典技巧

对于快速求链上最大值,我们可以建出这棵树的笛卡尔树,然后求笛卡尔树上两点的LCA,通过预处理ST表我们可以做到 $O(n \log n) - O(1)$

对于快速分裂一条链,我们发现其实只需要快速回答树上的一个祖先 x 到后代 y 上经过的第一个非 x 节点是谁,这可以通过类似 dsu 的思路,我们记忆化 x 到其所有轻儿子的答案,这样只需再考虑 y 是否在 x 的重儿子内,这样可以 $O(n \log n) - O(1)$

最后考虑到堆的时间复杂度和 set 的维护部分,总的时间复杂度为 $O(n \log n + Qm \log V \log(m \log V))$,可以通过本题

2.8 一些实现技巧

为了加快程序运行的速度,笔者实现的代码并不是求出来所有的 $m \log V$ 个数之后再进行后面的求值,而是同时维护了堆和 set

对于记忆化轻儿子来说,可以使用重儿子提前的 dfs 序,并动态分配内存

对于 rmq 问题来说,考虑到访问数组的时间,合理地设置数组的维度顺序可以显著地优化程序

3 总结

在测试点设置和命题方面,分数分布非常有梯度,部分分各式各样,能充分地让选手感受不同性质的子任务的差异,以及此题的模型特点,对于数据的生成,由于需要使用一个 $\log(m \log V)$ 的算法卡掉一个 $\log V$ 的算法,笔者想到的方法是对于树的结构和权值的分布进行特殊设置,方法大致如下:

对于树的结构,第 i 个点选择从随机前5层的节点连边,并按照对数级概率选取更靠前的层,并对后一半的点全部对应地连向前一半的点,这样做,树的度数,深度方面比较均匀,又可以卡掉随机剖分等等错误算法

对于权值的分布,笔者想到决定含有每一位的数的个数,并将这一位均匀分布

在树上,因此笔者选取了一个位大小到个数的函数,其呈现按照某种多项式地降低的特征

而从题目本身来说,从一个非常简洁的模型出发,经过一些对算法深入的思考,考察了选手推导性质,提出算法的能力,结合一系列经典简洁的方法,也考验选手的实现能力

实际上,大多数同学平时就会遇到一些位运算的模型,而最大值和与的结合似乎非常简洁但是又从未出现过,应该能给不少同学留下非常深刻的印象,同时笔者认为这题应当还有可以扩展的空间,是否可能去掉 $\log(m \log V)$,或者优化 $m \log V$ 这一上界,也希望对此方面感兴趣的同学继续深入思考,提出新的方法优化或者扩展这道题

4 验题

使用暴力检测了范围较小的`subtask`

集训队员郭宇豪在尝试解决子任务7时,提出了一种基于复杂度平衡的做法,考虑每 B 位贪心一次,我们需要知道一种前 B 位的数是否能被与出,那么我们考虑高维后缀和,可以做到 $O(B \cdot 2^B)$,每一次需要一次 $O(\log n)$ 的统计,共计 $\frac{\log V}{B}$ 次,根据复杂度的平衡,我们可以做到 $O(n + Q \frac{\log V \log n}{W(\log n \ln 2)})$,其中 W 表示朗伯 W 函数,在观察到性质后,子任务8也可以这样通过,在 $m, \log V$ 较小时,并无差异

5 参考资料

[1] 周镇东.浅谈一类树上路径相关问题,IOI2021 中国国家候选队论文集

[2] Wikipedia contributors. Bitwise operation. Wikipedia,https://en.wikipedia.org/wiki/Bitwise_operation#AND