基础优化算法 2

洛谷省选计划 B 组讲课

Gemini 张宸睿
qq：2367411769

洛谷网校

2024 年 7 月 18 日

dfs 和 bfs

- 搜索是对解集合的穷举，在算竞中一般没有办法时的方法，合理的写法以及剪枝对于搜索能得到的分数则至关重要。对于这一类算法的训练不能忽视（可以在平常打暴力训练）。
- 深度优先搜索：基于栈的搜索，与图论结合较深（dfs 序）。具体来讲实现一个递归函数，先往下一层的节点搜索，最后回溯到这个节点再换到同层的其他节点。
- 广度优先搜索：基于队列的搜索，这时就是按照”深度“进行搜索，按层便利的。这在做最短路问题时经常需要。

01 最短路

- 在一个有边权为 1 或 0 的图上，求出单源最短路。要求时间复杂度 $O(n+m)$。

01 最短路

- 普通最短路（bfs）可行的原因是它从队列中取出时已经时最短的了，可如果在 0-1 图上仍然使用之前的方法，则可能会出错。
- 改变一下做法，把 0 边扩展放入队首，1 边扩展放入队尾，这样就不会出问题了。而复杂度仍然为 bfs 的复杂度。只会入队 $O(m)$ 次，扩展 $O(n)$ 次。

记忆化

- 可以说是很类似 dp 了。在 dfs 的时候经常会经过同样的节点，这时后面的状态就全部相同了，那不如使用空间换时间，把这些状态记录下来就可以减少总的枚举次数。
- 记搜多数时候是 dp 的一种更好写的实现方式。

NOIP2017 逛公园

- 给定一个 n 个点 m 条边的有向带权图，求从 1 到 n 的长度不超过最短路 $+k$ 的路径条数，对大质数取模。
- $1 \leq n, m \leq 2 \times 10^5, 1 \leq k \leq 50$.

NOIP2017 逛公园

- 注意到这个 k 很小，定义状态为 $f(u,i)$ 代表从 $1 \to u$ 路径长度为 $dis(u)+i$ 的方案数，其中 $dis(u)$ 为从 $1 \to u$ 的最短路。
- 考虑转移 $f(v,i)$ 假设能从 $f(u,j)$ 转移，需要满足的就是 $dis(v)+i=dis(u)+j+w$，接下来就是暴力 dfs，然后记忆化即可，复杂度 $O((n+m)k)$。
- 这道题还有一个需要注意的细节是 0 环的处理。

Meet in the Middle

- 又名折半搜索。就是将原有的数据分成两部分分别进行搜索，最后在中间合并的算法。
- 显然，这需要我们的信息是可以单次合并的。
- 该算法可以使复杂度开根号。一般来讲这样的题都是一些指数复杂度，但是又开根之后可以接受。
- 一个著名的应用是 bsgs 算法，这里不过多扩展。

CF1779H Olympic Team Building

- n 个人进行比赛，第 i 个人的实力为 s_i。n 为 2 的幂。比赛将一直进行到只剩一个人——这个人为赢家。对于每轮比赛：设当前有 m 人。你要将这 m 人分为人数相等的两队。队伍的实力为每个人的实力总和。如果两支队伍实力相等，你会选择谁获胜；否则，实力大的获胜。输的队伍中的每个人都被淘汰。问每个人能否可能成为赢家。
- $4 \leq n \leq 32, 1 \leq s_i \leq 10^{15}$

CF1779H Olympic Team Building

- 首先不难想到对于每个人可以倒着考虑，然后不断把集合变大。一个很容易想到的想法是每次加上能加的最大的集合。可以尝试一下随机扰动+这个贪心做法。
- 但是看到这个数据范围我们再仔细思考一下，当集合大小为 16 时可以 $O(1)$ 判断，而当集合大小为 $1, 8$ 的时候贪心是对的。
- 剩下的部分，考虑暴力操作，但是这样状态数爆炸。改良一下这个贪心，变成如果集合 A 偏序集合 B 那么 A 一定比 B 优。这时在暴力枚举的过程中，我们只需要枚举没有被偏序的部分即可。

剪枝

- 剪枝是在搜索的过程中减去不需要的部分的做法，通常来讲剪枝的复杂度可以说玄学，但也有部分算法复杂度上界可以证明。无论是在正解还是暴力骗分中剪枝都是非常好用的做法。
- 可行性剪枝：如果当前分支已经和题目要求的不同那么就直接忽略。
- 最优化剪枝：如果当前分支没有之前已经搜索过的优，则不再往下搜索。
- 冗余性剪枝：当几个枝桠具有完全相同的效果的时候，只选择其中一个走就可以了。
- 顺序剪枝：结合了上面几种剪枝，不同的搜索顺序就可以有天差之别。如果是再特殊构造的数据中，将数据随机打乱可以获得不错的效果。另外将搜索算法与贪心结合，也可能得到很好的效率。

启发式搜索

- 迭代加深：把 dfs 变成 bfs，先设定一个最大深度，跑到这里就先不跑了。
- 设计估价函数，$g(x)$ 代表从起点跑到 s 的代价，$h(x)$ 代表从 x 跑到终点的估价，令 $f(x) = g(x) + h(x)$，从 f 中找到最小的一个跑，如果没有合法的 f 就不跑了。
- A* 是用 bfs 的方式，而 IDA* 则是用迭代加深的方式。

CF163D

- 给定长方体的体积 V，试求出三条边的长度，且使表面积最小。$1 \leq V \leq 10^{18}$.

CF163D

- 记三边为 a, b, c，且不妨设 $a \le b \le c$。那么一定有 $a \le \sqrt[3]{V}$。
- 暴力枚举 a，即枚举所有 V 的 $\le \sqrt[3]{V}$ 的约数。此时已知 $bc = \dfrac{V}{a}$，只需要求出 $a(b+c)$ 的最小值即可。
- 由均值不等式 $(b+c) \ge 2\sqrt{bc}$，当 $b = c$ 的时候取等。
- 考虑当前有一个答案 ans，如果在均值取等的情况下仍然比 ans 大，那此时的 a 直接剪枝即可。
- 否则直接，暴力搜索 b，其实搜索 ab，和搜索 c 的复杂度相同，显然是 $O(\sqrt{V})$。加上剪枝，以及约数并卡不满等优化，可以通过。

NOI2003 智破连环阵

- 给定一个二分图，两个点集大小分别为 n, m。一个左部点可以匹配连续的一段右部点。
- 求最少需要几个左部点，匹配完右边全部点。
- $1 \leq n, m \leq 100$。每个左部点期望与右边连边 3 条。

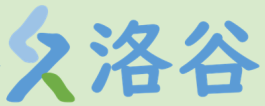
NOI2003 智破连环阵

- 题意已经是转化过的了，想看原题的可以自行搜索+思考。
- 将右边的区间当作点，这样就是正常的二分图匹配了。
- 搜索每种区间划分的方式，跑二分图匹配看看是不是完美匹配。
- 按从多到少的划分方式枚举，并且算上估价函数 $h(i)$ 代表从 i 开始往后匹配，最少能划分几个区间。用 A* 加速可以通过此题。

离散化

- 当我们的做法，只依赖与数据间的偏序关系时，我们就可以对数据进行离散化。
- 具体来说，我们对每个值进行一个映射，并且映射后，数据的偏序关系不变。
- 显然，对于 $O(n)$ 的数据，离散化后值域也是 $O(n)$ 的。

扫描线



- 很多在看题解的时候，就会看到“扫描线一下即可”这样的说法。可扫描线这个算法究竟是什么？
- 我在第一次接触扫描线的时候是求矩形面积并（当然还有各种形状），然而扫描线的意义不止于此。
- 扫描线，其实是一种降维技巧。
- 对于一个高维立方体，不妨设每一维都是 $1 - side$ 的。对其中一维按偏序关系从前往后遍历，记录历史的其他维度的信息。
- 这里用“线”这个字还算形象，因为我们解决的大部分问题都是二维的，那剩下的维度就是一条线。

扫描线

- 由于今天不讲数据结构，所以我不想过多浪费时间在 ds 技巧上。

NOIP2023 天天爱打卡

- 重温一下这个题。
- 一共有 n 天，大 Y 可以花费 d 的能量在任何一天跑步，但大 Y 不能在连续的 k 天跑步。
- 此外，将给出 m 段任务区间 $[l_i, r_i]$。如果在这段区间的每一天都跑了步，则会获得 v_i的能量。
- 大 Y 的初始能量为 0，请你最大化跑完步后大 Y 最终的能量。

NOIP2023 天天爱打卡

洛谷

- 考虑暴力 dp，f_i 代表第 i 天跑步，前 i 天的最大能量。同时要记录一下，前 i 的答案，就是不需要第 i 天强制跑步。
- 转移也很简单，直接枚举上一次不跑步的时候，然后看这一段跑步的时间包含了哪些区间，这个可以扫描线做。
- 这个做法甚至是 $O(V^2)$ 的，如果一段时间没有任务，那么干嘛要跑步。所以要跑的天是非常少的。
- 直觉告诉我们，只需要保留 l, r 周围的点即可，将这些点离散化跑 dp。然后转移线段树优化，复杂度 $O(n \log n)$。
- 可事实上这题卡常，对于我的做法，只用保留 $l, r, l-1$。因为我如果要休息，一定是在一个 $l-1$ 的地方休息。

区间数颜色

- 可能是这个题：
  https://www.luogu.com.cn/problem/P1972

区间数颜色

- 维护这个颜色上一次出现的位置 pre_i，只在每个颜色在区间内第一次出现的时候统计答案。问题就转变为区间统计 $\leq l-1$ 的数的个数。这东西是可减的，容易容斥成前缀 $\leq x$ 的数的个数，就变成了二维数点。
- 这个你可以可持久化，也可以离线扫描线。

未知来源题

- 给定一棵 n 个点的树，有 q 次查询，每次查询区间 $[l, r]$，表示询问当这个树仅剩点和边的编号在 $[l, r]$ 之间时连通块的个数。
- $1 \leq n \leq 10^6$.

未知来源题

- 答案为点数减去端点均在 $[l, r]$ 内的边数。
- 对于一条边 (u, v)，不妨设 $u < v$。
- 对于每条边，将其双射到点 (u, v) 上，对于询问 $[l, r]$，即为矩形 $(l, l) - (r, r)$ 数点。

未知来源题

- 给定很多模式字符串，每次查询时给两个字符串 s_1, s_2，问有多少模式字符串前缀是 s_1，后缀是 s_2

未知来源题

- 考虑分别按正序和倒序建 trie，然后每次查询的就是 trie 树上的一个子树。
- 转换成区间，整个查询就是一个矩形，问题又变成成了二维数点。

未知来源题

- 给定一个序列，每次查询区间中出现偶数次的数的异或和。

未知来源题

- 如果是奇数次的数，直接异或即可。
- 全部增加一次，即可求出偶数的。现在问题变成了每种颜色只增加一次。
- 类似区间数颜色的题，也就是把数个数变成了异或和，大同小异。

扫描线

- 还有一种扫描线一维是右端点，一维是左端点。
- 具体来说就是从左往右扫右端点，用 ds 维护每个点作为左端点的答案。

Luogu 8512

- 有一个操作序列(l_i, r_i, v_i)。
- 现在，有 q 个询问 l,r。每次询问，你初始有一个长度为 m 的序列 c，初值全是 0。
- 现在我们从 l 到 r 执行这 $r-l+1$ 个操作。每个操作是将 $c[l_i]$ $c[r_i]$ 赋值为 v_i。
- 询问所有操作结束后整个 c 的序列所有数的和。

Luogu 8512

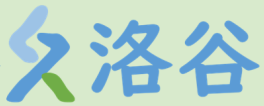
- 对于每次询问暴力修改，可以使用 ‘std::set‘ 维护连续段。
- 考虑对操作扫描线，并且在覆盖的同时记录时间信息。
- 对于一次查询的 l，只考虑这之后的区间操作的变化。
- 具体来讲，对于每一次修改的连续段，我们在它插入的时间处将其减去/加上。

推荐练习

- 可能需要一些 ds 技巧。
- 【模板】扫描线（矩形面积并）
- CF453E
- CF1824D

推荐练习

- THUPC2022 决赛 rsraogps（不推荐写 Ynoi 版）
- Ynoi Easy Round 2021 TEST152
- Luogu P7560
- NOIP2022 比赛



启发式合并

- 对于若干个集合 $S_1, S_2, \ldots S_m$，不妨令 $|S_1| \leq |S_2| \leq |S_m|$。
- 现在想要把它们合并成一个大集合，保留 S_m，将其他小的集合合并到那个集合里去。
- 这样子，假设将合并的过程形成一个树的结构，重链上的点，并不需要合并。
- 那么每个叶节点（原本的元素）合并的次数即为跳到根节点上轻边的条数。
- 按照轻重链剖分的理论，这最多 $O(\log n)$ 条边。

并查集按秩合并

- 也就是在合并的时候将小的集合挂到大的集合上，这样子查找根的时候，往上跳的步数是 $O(\log n)$ 的。

平衡树的启发式合并

- 结论：直接把小的平衡树里的点塞到最大的里面，这样子总的复杂度是 $O(\log n)$。
- 前提是你用了 finger search，但是如果不用差别不大。很少人会卡

Luogu P9067

- 给定一棵 n 个节点的树，第 i 个点有点权 a_i。
- 定义一个点 x 所在的极大同色连通块为一个**极大的**点集 S，满足 $x \in S$，且对任意点集中的元素 i, j，可以找到一个节点序列 $p_1, p_2, \ldots p_t$，满足 $p_1 = i$，$p_t = j$，且对任意 k 为 $[1, t)$ 中的整数，满足 p_k 与 p_{k+1} 在树上相邻，且 $a_{p_k} = a_{p_{k+1}}$，且 $p_k \in S$。
- 有 m 次操作：
- ‘1 x y‘：给出一个点 x，将其所在的极大同色连通块中每个点的点权修改为 y。
- ‘2 x‘：给出一个点 x，查询其所在的极大同色连通块的大小。

Luogu P9067

- 类似颜色段，合并至多发生 $O(n)$ 次，但是与颜色段不同的是，检查是否需要合并不再是 $O(1)$ 的了。
- 考虑把所有，在 x 的连通块下方（直接相连）的颜色为 y 的连通块记录下来。
- 父亲的部分就是 $O(1)$ 讨论。
- 这个记录的东西，类似于一个桶，那在合并的时候就需要用启发式合并。
- 参考题解区 map 套 vector 的写法非常好写。

树上启发式合并

- 一种用于解决树上点对问题的做法。
- 对于一棵树，考虑只遍历所有的轻子树，计算贡献。重子树无论是从下网上继承，还是加入都不重新遍历。
- 容易发现这样复杂度和启发式合并相同，均为 $O(n \log n)$。

省选联考 2020 A 卷树

- 给定一棵 n 个结点的有根树 T，结点从 1 开始编号，根结点为 1 号结点，每个结点有一个正整数权值 v_i。
- 设 x 号结点的子树内（包含 x 自身）的所有结点编号为 $c_1, c_2, \ldots, c_k$，定义 x 的价值为：
  $val(x) = (v_{c_1} + d(c_1, x)) \oplus (v_{c_2} + d(c_2, x)) \oplus \cdots \oplus (v_{c_k} + d(c_k, x))$
- 其中 $d(x, y)$ 表示树上 x 号结点与 y 号结点间唯一简单路径所包含的边数，$d(x, x) = 0$。$\oplus$ 表示异或运算。
- 请你求出 $\sum_{i=1}^{n} val(i)$ 的结果。

省选联考 2020 A 卷树

- 这道题主流做法是 01 trie，可是对于不会这个科技的同学也可以尝试 dsu on tree。
- 考虑加 1 带来的结果，应该是末尾连续的一段 1 消失，前面最高位的前一位加 1。
- 所以维护这个每一种 1 的个数，即 $cnt_{u,k}$ 代表 u 子树内最后一段 1 是 $2^k - 1$ 的个数，单纯的维护和求答案是 $O(\log V)$。
- 合并这个东西用树上启发式合并，总的复杂度为 $O(n \log n \log V)$。
- bonus：套上 01 trie 的皮不知道能不能做到 $O(n(\log n + \log V))$ 的复杂度。

推荐练习

- luogu P5290
- CF741D
- CF600E
- P9168 的 48 分

启发式分裂

- 启发式合并的反向过程，在分治的时候用小的那边计算大的那边的贡献，并且继续递归下去。
- 递归的层数虽然可能很多，但是这其实就是启发式合并的逆过程，所以复杂度还是 $O(n \log n)$。

未知来源题

- 有 n 个奖杯，第 i 个奖杯的颜色是 c_i。
- 还一个长度为 n 的数组 f，并且保证这个数列单调递减。
- 一个长度为 l 的奖杯区间是好的，当且仅当这个区间中的所有颜色的出现次数均不小于 f_l。
- 现在他想要找到最长的好的区间，他让你来帮他完成这个任务。

未知来源题

- 直接扫描线并不是很好做因为 f 的变化，先考虑一个暴力的做法。
- 假设当前区间长度为 L，删掉所有出现次数不到 L 的数，然后递归到每个小区间里去做。
- 考虑当前区间为 $[l, r]$，并且 pos 是第一个要被删的地方。直接分成 $[l, pos-1], [pos+1, r]$ 两个区间。
- 然而现在不知道 pos，需要暴力扫一遍区间。
- 事实上，只需要枚举 $[l, pos]$ 或者 $[pos, r]$ 一边即可，那这个时候，选择小的那一边遍历，复杂度即为 $O(n \log n)$。

推荐练习

- luogu P4755
- CF1156E
- ATabc282Ex
- 这三个题都是笛卡尔树启发式分裂的经典代表。

Thanks!