

DFS BFS

冉雨杭

2023 年 8 月 4 日

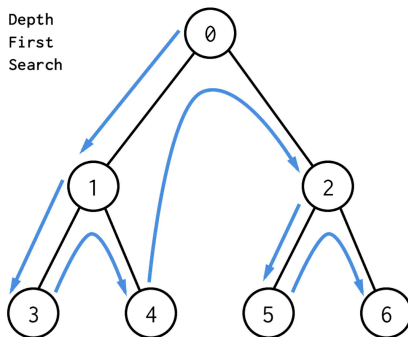
- 搜索：利用计算机的高性能来有目的的穷举一个问题解空间的部分或所有的可能情况，从而求出问题的解的一种方法
- 搜索需要按照一定的顺序，保证能够把所有情况都穷举到，不能盲目进行搜索
- 根据不同问题的类型采用合适的搜索方式

- 例子：求所有满足 $a_1 + a_2 + a_3 = 50$ 的正整数解
- 这个我会！三重 for 循环轻松搞定！
- 要是变成求所有 $a_1 + a_2 + \dots + a_n = 50$ 的正整数解呢？

- 我会，写 n 重循环！（ n 都不是固定的你怎么写？）
- 使用搜索来寻找所有满足 $a_1 + a_2 + \cdots + a_n = 50$ ，其中 $(1 \leq a_1 \leq a_2 \leq a_3 \cdots \leq a_n)$ 的答案

深度优先搜索 (DFS)

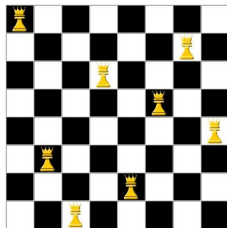
- DFS: **先递归，后回溯**，首先一路走到底，直到不能再下就回溯到上一个能继续往下走的地方，直到搜索完成



- 如果当前到达了结束状态，则返回
- 枚举当前的选择是什么，并打上相应的标记，递归进入下一个状态
- 递归结束后记得**清空所有标记**

八皇后问题

- 问题描述：在 8×8 格的国际象棋上摆放八个皇后，使其不能互相攻击，即任意两个皇后都不能处于同一行、同一列或同一斜线上，问一共有多少种摆法？
- 我会 8 重循环！枚举每个 8 个棋子分别放在哪个格子！
- 时间复杂度太高： $\binom{64}{8} = 4426165368$
- 假设第一个和第二个皇后冲突了，剩下的就没必要再枚举了！



八皇后问题

- 可以枚举阶乘，这样复杂度变成了 $8! = 40320$
- 再一一检验对角线是否满足即可
- 一个更好的方法是 dfs 的过程中直接检查，这样如果对角线在比较早的地方就不满足了可以剪枝

- 给定一个 $m \times m$ 方格的迷宫，迷宫里有 k 处障碍，障碍处不可通过。在迷宫中移动有上下左右四种方式，每次只能移动一个方格。数据保证起点上没有障碍。给定起点坐标和终点坐标，每个方格最多经过一次，问有多少种从起点坐标到终点坐标的方案
- 数据范围： $1 \leq n, m \leq 5, 1 \leq k \leq 10$

- 从 n 个整数 x_1, x_2, \dots, x_n 中选 k 个相加，问有多少种方案得到的和是质数？
- 数据范围： $1 \leq n \leq 20, k \leq n, 1 \leq x_i \leq 5 \times 10^6$
- 提示：质数可以使用根号复杂度暴力判断

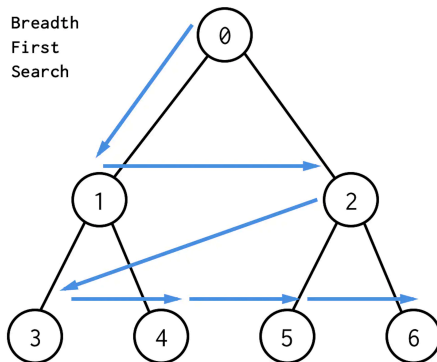
- 给一个 $n \times n$ 的方格矩阵，还有 $n \times n$ 个整数，让你将这些整数填入矩阵，使得每行每列每个对角线上整数的和都相等
- 数据范围： $1 \leq n \leq 4$, $-10^8 \leq a_i \leq 10^8$

P1406 方格填数

- 直接 dfs 进行搜索
- 剪枝：每一行，每一列，对角线必须都是总和的 $\frac{1}{n}$

广度优先搜索 (BFS)

- BFS: 逐渐往外拓展搜索, 每次往外延伸一层
- BFS 常用来解决图上边权只包含 0 和 1 的最短路问题



- 将初始状态放入队列中，并打上标记
- 每次从队列中取出一个状态，遍历它能到达的所有其它状态，将这些状态中没有访问过的状态打上访问的标记，并放入队列中
- 重复上述操作直到队列为空
- 复杂度： $O(n + m)$ ，其中 n 是状态数， m 是总的转移数

图上最短路

- 有一个 n 个点, m 条边点的图, 边权都是 1。给定起点 s 和终点 t , 问从 s 到 t 的最短路是多少?
- 数据范围: $1 \leq n \leq 10^6, 1 \leq s, t \leq n$

单词接龙

描述

给出两个单词（`start` 和 `end`）和一个字典，找出从 `start` 到 `end` 的最短转换序列，输出最短序列的长度。

变换规则如下：

1. 每次只能改变一个字母。
2. 变换过程中的中间单词必须在字典中出现。（起始单词和结束单词不需要出现在字典中）

- ① 如果不存在这样的转换序列，返回 0。
- 所有单词具有相同的长度。
- 所有单词只由小写字母组成。
- 字典中不存在重复的单词。
- 你可以假设 `beginWord` 和 `endWord` 是非空的，且二者不相同。
- $\text{len}(\text{dict}) \leq 5000, \text{len}(\text{start}) < 5$

单词接龙

- 将单词抽象成图上的点，转移抽象成图上的边
- 变成了图上的最短路问题

- 给定一个 n 个点 m 条边的无向图，有一个初始起点被染色，一个点下一轮会被染色当且仅当它周围有点当前轮被染色。 q 次询问，每次询问一个点在某个时刻是否被染色
- 数据范围： $1 \leq n, m, q \leq 10^5$

- 发现 L 时刻如果能被染色, $L + 2, L + 4 \dots$ 都会被染色
- 只跟时刻奇偶的最短路有关
- 多一维表示奇偶即可

- 你有一个 4 位的密码锁 (包含数字 0-9)，每次操作可以选取一段连续的区间往上或者往下转动一次，注意每一位都是环状的，一共给 T 次询问，每次问从一个 4 位密码到另一个 4 位密码最少要多少次操作？
- 数据范围： $1 \leq T \leq 10^5$
- 出处： <https://codeforces.com/gym/103427/problem/J>

- 可以把每一种密码看成一个状态，则一共有 10^4 种状态
- 考虑预处理每一种状态到其它所有状态的路径，复杂度是 $10^8 \times C$ ，其中 C 是转移的复杂度
- 上述复杂度太高，不能通过该题，发现跟具体是什么数字无关（各个数字没有本质区别），只跟初始状态和终止状态之间的偏移量有关
- 于是可以只预处理从 $(0, 0, 0, 0)$ 出发到其它任何状态的最短路即可，复杂度大大降低

Caught in a maze

- 给定 n 个点, m 条边的连通图, 每次两个人会同时选择一条边进行移动, 现在给 q 次询问, 每次两个人的起点 (x, y) , 问至少需要多少次操作两个人能在同一个点相遇?
- $1 \leq n, m \leq 1500, q \leq 2 \times 10^5$

Caught in a maze

- 考虑到这从所有点 (i, i) 出发到所有 (x, y) 的最短路
- 每次枚举两个人走的边
- 复杂度是三次方的
- 这样会 TLE, 因为需要同时枚举两个人走的边

- 考虑多一维 0/1, 表示当前该谁走, 这样每次转移只用枚举一个人的转移边
- 复杂度优化到 $O(n(n + m))$

- 有一个 n 个点, m 条边点的图, 边权为 0 或者 1。给定起点 s 和终点 t , 问从 s 到 t 的最短路是多少
- 数据范围: $1 \leq n \leq 10^6, 1 \leq s, t \leq n$

- 将初始状态放入队列中，并打上标记
- 每次从队列中取出一个状态，遍历它能到达的所有其它状态，将这些状态中没有访问过的状态打上访问的标记，如果该边边权为 0，则将其塞到队首；如果该边边权为 1，则将其塞到队尾。
- 重复上述操作直到队列为空
- 发现我们的队列需要支持：在最前面加入一个数 (`push_front`)，在最后面加一个数 (`push_back`)，取出最前面的数 (`pop_front`)。于是使用双端队列即可 (STL 中的 `deque`)

```
void solve(int s){
    queue<int> q;
    q.emplace(s);
    vector<int> vis(n);
    while(!q.empty()) {
        int u = q.front();
        q.pop();
        for (auto [v, c] : adj[u]) {
            if(vis[v]) continue;
            vis[v] = 1;
            if(c == 0) {
                q.push_front(v);
            }
            if (c == 1) {
                q.push_back(v);
            }
        }
    }
}
```

- 从起点走到终点， $n \times m$ 的网格，每个位置有一个小写字母，若 $s[x][y] = s[nx][ny]$ ，则移动的花费为 0，否则花费为 1，求花费最少？
- 数据范围： $1 \leq n, m \leq 2000$

- 01BFS 即可

- 给出一个数字 n , 和 m 个一位数字, 求 n 的最小倍数, 使得其中不会出现这 m 个数字
- 数据范围: $1 \leq n \leq 10^4$

- 因为需要找到最小，如果使用 dfs 则需要枚举完所有情况，在这类问题上显然没有 bfs 优秀
- 使用 bfs，每次枚举在后面加哪个数字的时候按从小到大顺序，即可保证出来的数字是从小到大的
- 这样会超时，考虑优化？

- 假设所有数字对 n 取模，当取模结果为 0 的时候说明找到了答案
- 从前往后枚举加入的数字

- 假设当前数字模 n 结果为 x ，加入的数字为 y ，新的结果为 $(10x + y) \bmod n$
- 如果某一种余数出现过了，它俩是否是本质相同的，是不是只需要保留第一个，第二次及以后出现的就可以不用再管了
- 例子： $n = 3$, 1 和 4 是同余的，4 永远不可能成为答案首位

- 从 0 开始 bfs
- 每次枚举新加入的数字 y
- 算出加入后的新结果 $(10x + y) \bmod n$
- 如果余数是 0，输出答案
- 如果已经访问过，则跳过
- 否则加入队列中，打上访问标记

- 题意：给出 n 个点 m 条边的补图，求原图连通块个数及每个连通块的大小
- 数据范围： $1 \leq n, m \leq 2 \times 10^5$
- <https://codeforces.com/contest/920/problem/E>

- 直接按照正常 BFS 做，每次枚举 n 条边，复杂度 $O(n^2)$
- 不存在的边很少
- 只用维护当前没被访问过的点，然后遍历，复杂度好像是对的？

- 用链表或者 set 维护还没有被访问过的结点，每次队列取出一个点 u ，遍历所有未访问过的点 v ，如果有边则加入 v ，否则跳过
- 前者只会发生 n 次，后者只会发生 m 次
- 复杂度 $O(n + m)$ 或者 $O(n + m)\log n$

```

void bfs(int s) {
    queue<int> que;
    que.push(s);
    unused.erase(s);
    int t = 0;
    while (!que.empty()) {
        int u = que.front();
        que.pop();
        ++t;
        auto it = unused.begin();
        while (it != unused.end()) {
            int v = *it;
            auto nxt = next(it);
            if (!e[u].count(v)) {
                que.push(v);
                unused.erase(it);
            }
            it = nxt;
        }
    }
    sol.push_back(t);
};

```

习题

- P1406
- P1605
- P1443
- P2895
- P2036
- P1162
- P1825
- P5663
- cf1064D

谢谢!