

「あの日見た花の名前を
僕達はまだ知らない。」

搜索与剪枝

——59级信息组：马一凌

あの日見た
花の名前を
僕達はまだ
知らない。

我们要学的东西

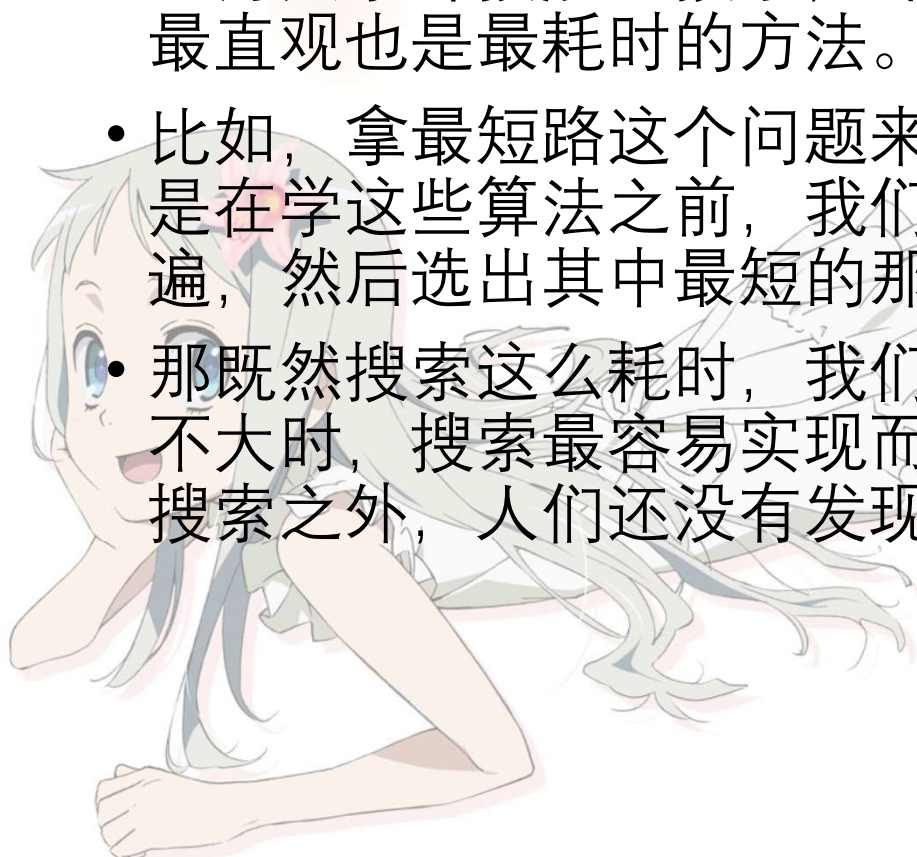
- 搜索是什么
- 搜索能解决什么问题
- 搜索的优化



あの日見た
花の名前を
僕達はまだ
知らない。

搜索是什么

- 因为大家都接触过搜索，那就简单来讲，搜索是解决一切问题的最直观也是最耗时的方法。
- 比如，拿最短路这个问题来说，我们已经学过了好几种算法，但是在学这些算法之前，我们能采用的方法就是把所有的路都走一遍，然后选出其中最短的那一条。
- 那既然搜索这么耗时，我们为什么还要学呢？首先，当数据规模不大时，搜索最容易实现而且正确性显然。其次，很多问题除了搜索之外，人们还没有发现更简单的方法。



あの日見た
花の名前を
僕達はまだ
知らない。

搜索能解决什么问题

- 首先说两道最简单最典型的搜索：
 数字三角形搜索版 codevs 3220
 过河卒codevs 1010
- 下面是一些简单题目



あの日見た
花の名前を
僕達はまだ
知らない。

例题1

- 某同学考试，在 $N \times M$ 的答题卡上写了A,B,C,D四种答案。他做完了，又不能交，一看表，离打铃还有N久。他开始玩一个游戏：选一个格子X,Y，从这个格子出发向4个方向找相同的选项，找到的再如此。求形成的图形的面积。（一个选项占一个单位面积）
- $N, M \leq 15$.



あの日見た
花の名前を
僕達はまだ
知らない。

例题1

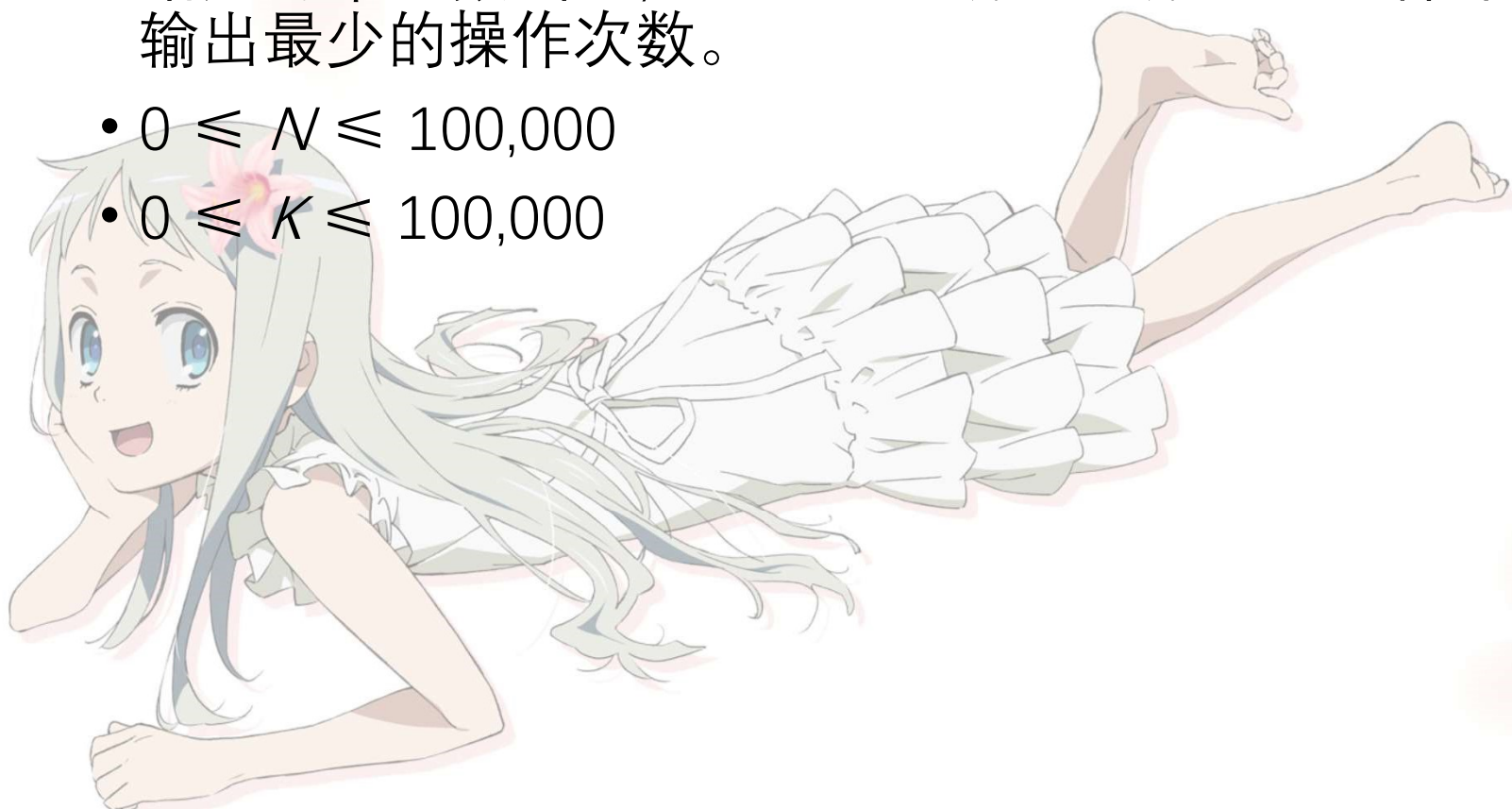
- 这个题是要求得与 (x, y) 相连的一个连通块
- 每次搜索时，如果 (x, y) 周围的四个格子中有和它答案相同的就向四周搜索。
- Codevs 2919



あの日見た
花の名前を
僕達はまだ
知らない。

例题2

- 给定两个整数 n 和 k ，通过 $n+1$ 或 $n-1$ 或 $n*2$ 这3种操作，使得 $n=k$ ，输出最少的操作次数。
- $0 \leq N \leq 100,000$
- $0 \leq K \leq 100,000$



あの日見た
花の名前を
僕達はまだ
知らない。

例题2

- 宽搜，对于一个 n ，每次将 $n+1$ ， $n-1$ ， $n*3$ 放入队列，并用一个bool数组标记一下，直到搜索到 k 。
- 注意搜索边界。
- Poj 3278



あの日見た
花の名前を
僕達はまだ
知らない。

例题3

- 在 3×3 的棋盘上，摆有八个棋子，每个棋子上标有1至8的某一数字。棋盘中留有一个空格，空格用0来表示。空格周围的棋子可以移到空格中。要求解的问题是：给出一种初始布局（初始状态）和目标布局（为了使题目简单，设目标状态为123804765），找到一种最少步骤的移动方法，实现从初始布局到目标布局的转变。
- （经典八数码）



あの日見た
花の名前を
僕達はまだ
知らない。

例题3

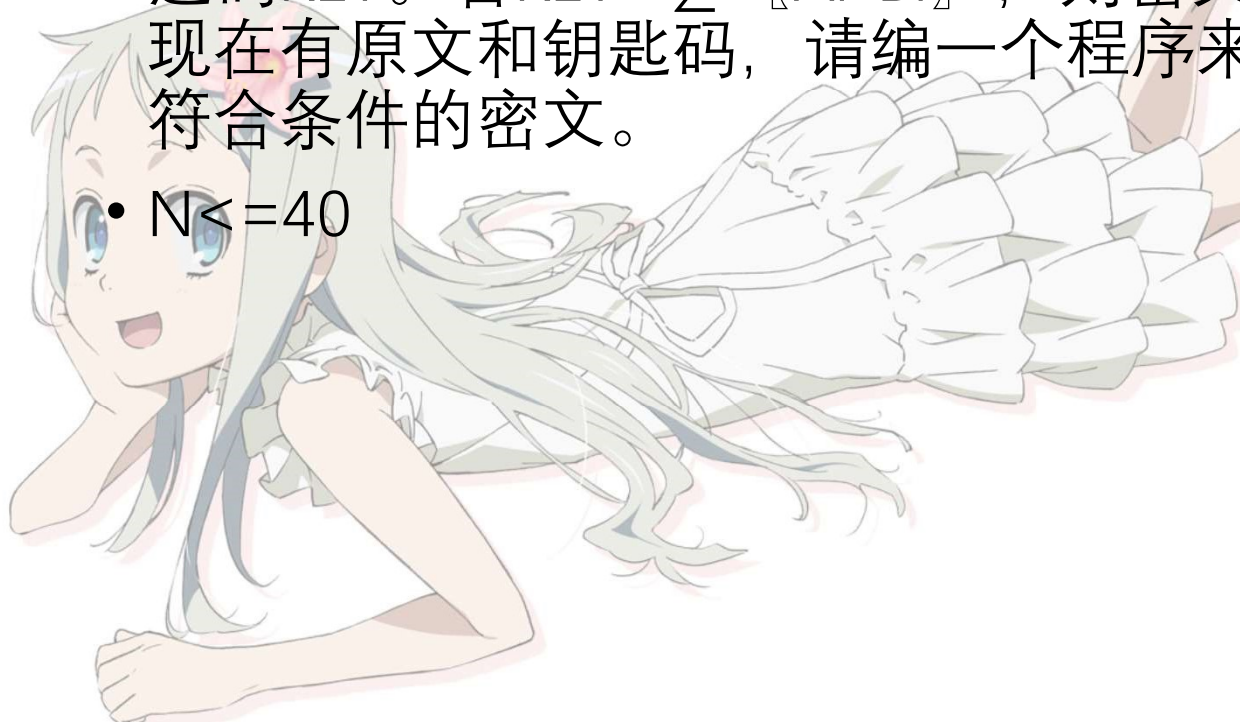
- 同样也是宽搜的思想，在队列中储存棋盘的状态，对于一个状态，每走一步就产生一个新状态，再放入队列中，直到搜索到最终状态。注意在搜索中要进行hash判重，这是解决问题的关键。
- Codevs 1225



あの日見た
花の名前を
僕達はまだ
知らない。

例题4

- 关于某种密码有如下描述：某种密码的原文A是由N个数字组成，而密文B是一个长度为N的01数串，原文和密文的关联在于一个钥匙码KEY。若 $KEY = \sum [A_i * B_i]$ ，则密文就是原文的一组合法密码。现在有原文和钥匙码，请编一个程序来帮助他统计到底有多少个符合条件的密文。
- $N \leq 40$



あの日見た
花の名前を
僕達はまだ
知らない。

例题4

- 这道题的暴力做法很显然，直接枚举每一位是0或1即可，这样的做法在数据范围是40的条件下显然不能通过。
- 可以先用暴力跑20位，用hash存储一下，然后跑另外20位得到答案。



あの日見た
花の名前を
僕達はまだ
知らない。

搜索的优化

- 通常的优化方式有：

剪枝

迭代加深搜索

记忆化搜索

双向搜索



あの日見た
花の名前を
僕達はまだ
知らない。

剪枝

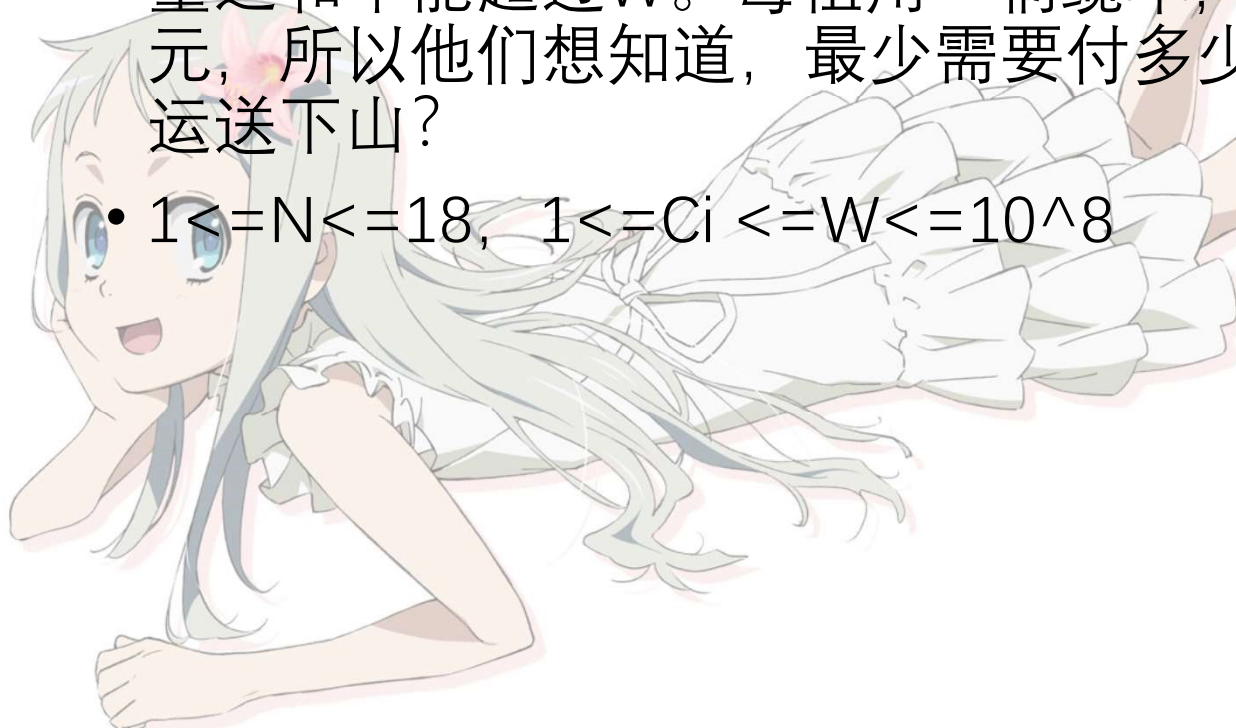
- “剪枝”，顾名思义，就是把枝条剪掉。也就是说在搜索的过程中，会有一些状态对最后的答案是不会产生贡献的，我们就把它扼杀在摇篮里，就会节省一些时间。



あの日見た
花の名前を
僕達はまだ
知らない。

例题5

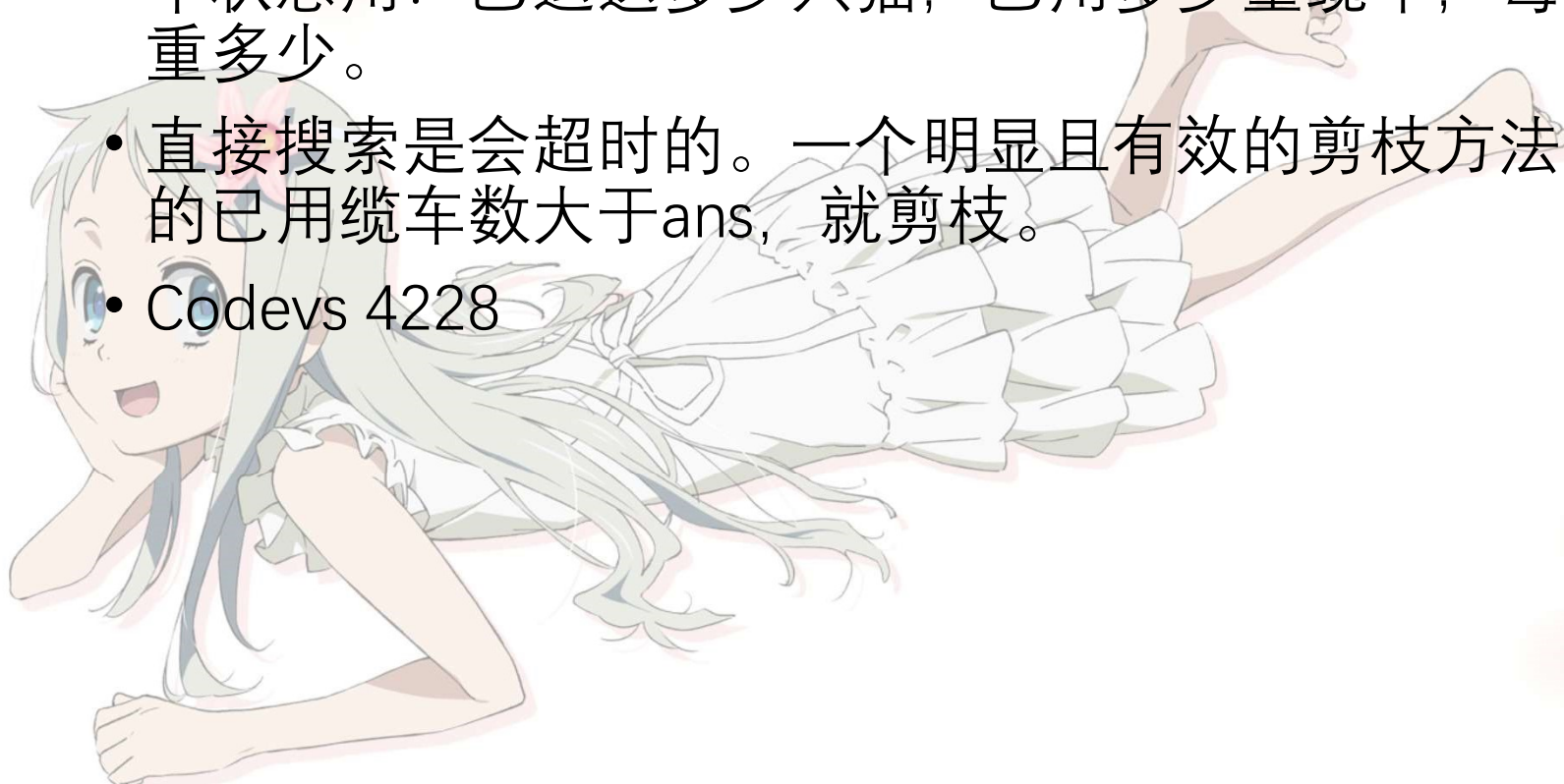
- 有N只小猫要坐缆车下山。索道上的缆车最大承重量为W，而N只小猫的重量分别是C1、C2……CN。当然，每辆缆车上的小猫的重量之和不能超过W。每租用一辆缆车，Freda和rainbow就要付1美元，所以他们想知道，最少需要付多少美元才能把这N只小猫都运送下山？
- $1 \leq N \leq 18$, $1 \leq C_i \leq W \leq 10^8$



あの日見た
花の名前を
僕達はまだ
知らない。

例题5

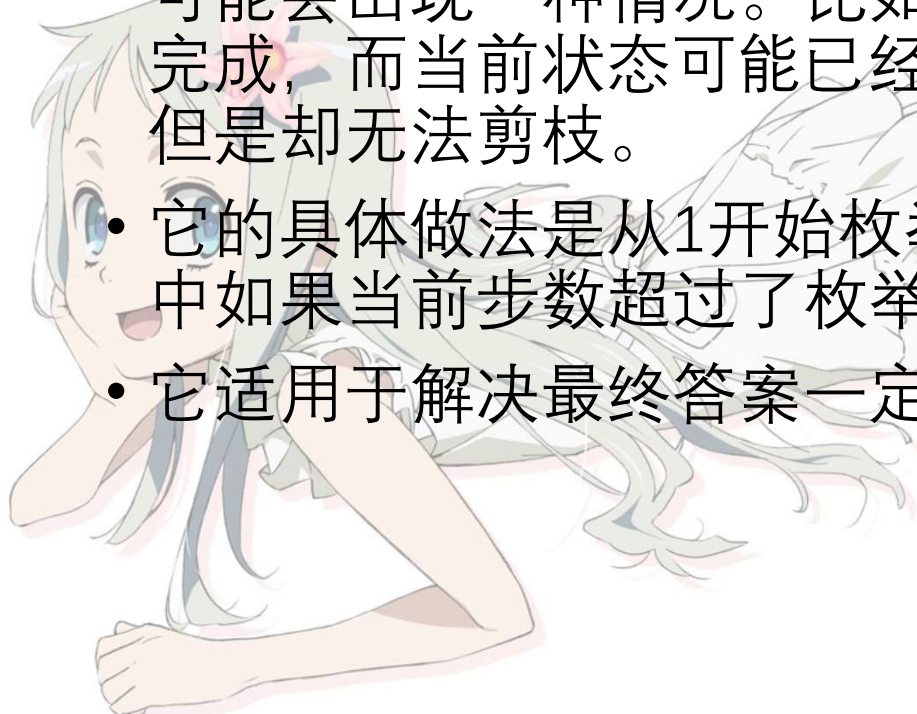
- 对于每只猫，要么用已有的缆车，要么新建一辆车。所以描述一个状态用：已运送多少只猫，已用多少量缆车，每辆已用的车载重多少。
- 直接搜索是会超时的。一个明显且有效的剪枝方法是当当前状态的已用缆车数大于ans，就剪枝。
- Codevs 4228



あの日見た
花の名前を
僕達はまだ
知らない。

迭代加深搜索

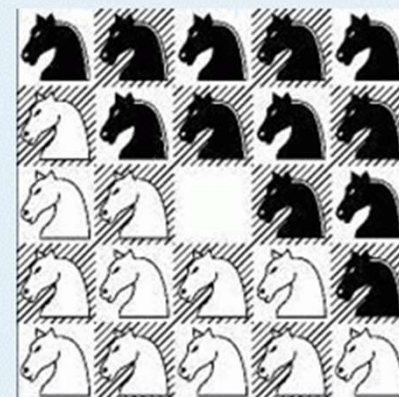
- 什么是迭代呢？它是一种优化dfs的算法。
- 众所周知，dfs是一种一搜搜到底的算法，但是在搜索的过程中，可能会出现一种情况。比如说，我们要求得一个问题最少几步能完成，而当前状态可能已经超过了这个步数却没有走到最终答案，但是却无法剪枝。
- 它的具体做法是从1开始枚举几步能完成，每次都dfs，在dfs过程中如果当前步数超过了枚举的步数，就剪枝。
- 它适用于解决最终答案一定很小的题目。



あの日見た
花の名前を
僕達はまだ
知らない。

例题6

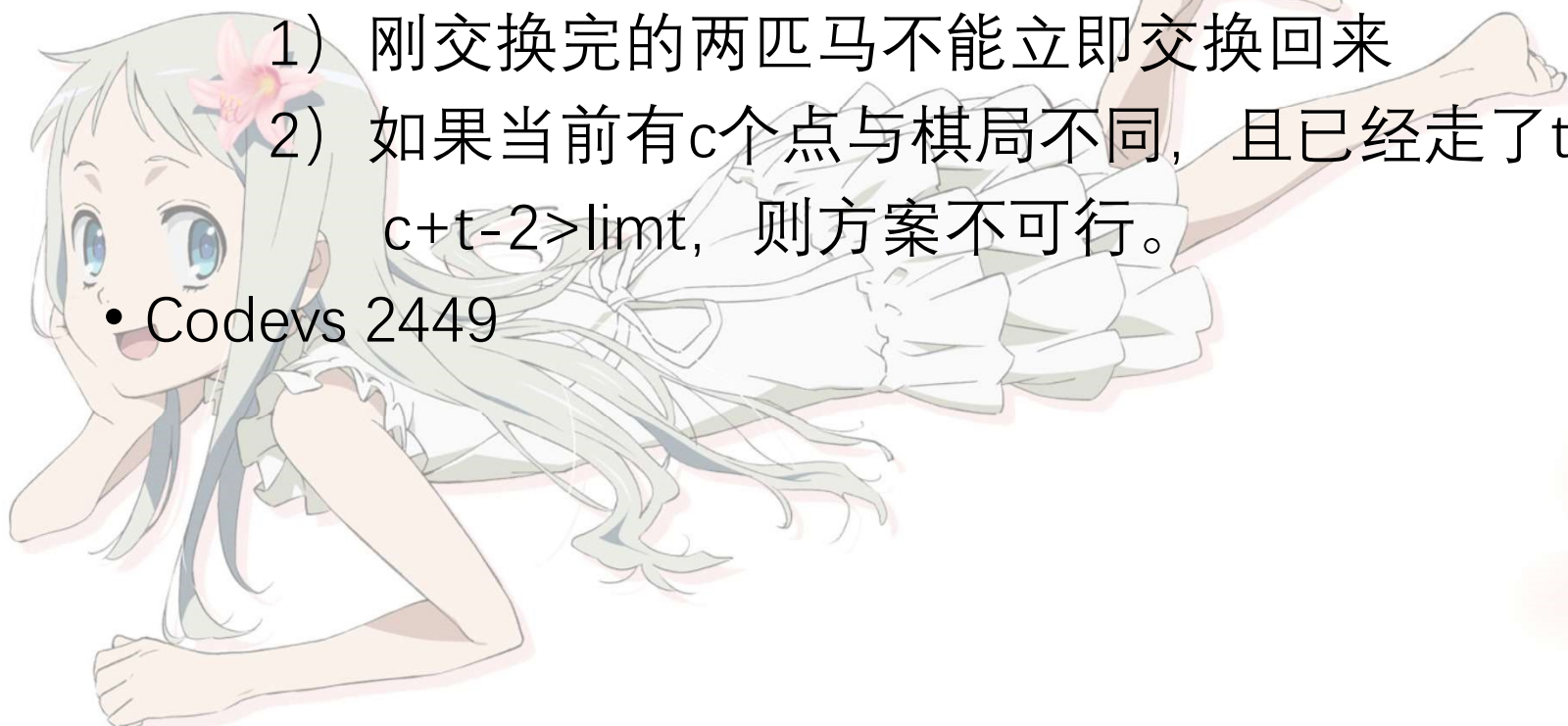
- 在一个 5×5 的棋盘上有12个白色的骑士和12个黑色的骑士，且有一个空位。在任何时候一个骑士都能按照骑士的走法（它可以走到和它横坐标相差为1，纵坐标相差为2或者横坐标相差为2，纵坐标相差为1的格子）移动到空位上。
- 给定一个初始的棋盘，怎样才能经过移动变成如下目标棋盘：
- 为了体现出骑士精神，他们必须以最少的步数完成任务。如果能在15步以内（包括15步）到达目标状态，则输出步数，否则输出-1。



あの日見た
花の名前を
僕達はまだ
知らない。

例题6

- 这个题目就很符合迭代加深搜索的定义了。
- 除此之外，还要加剪枝。我加了以下两个剪枝
 - 1) 刚交换完的两匹马不能立即交换回来
 - 2) 如果当前有 c 个点与棋局不同，且已经走了 $t-1$ 步，如果 $c+t-2>limt$ ，则方案不可行。
- Codevs 2449



あの日見た
花の名前を
僕達はまだ
知らない。

记忆化搜索

- 记忆化搜索就是在递归中先记录，后返回；再次递归时查阅记录，如果记录中有，则直接返回。
- 记忆化搜索是连接搜索与dp的桥梁，大部分能写成记忆化搜索的题目都可以用递推来写。



あの日見た
花の名前を
僕達はまだ
知らない。

例题7

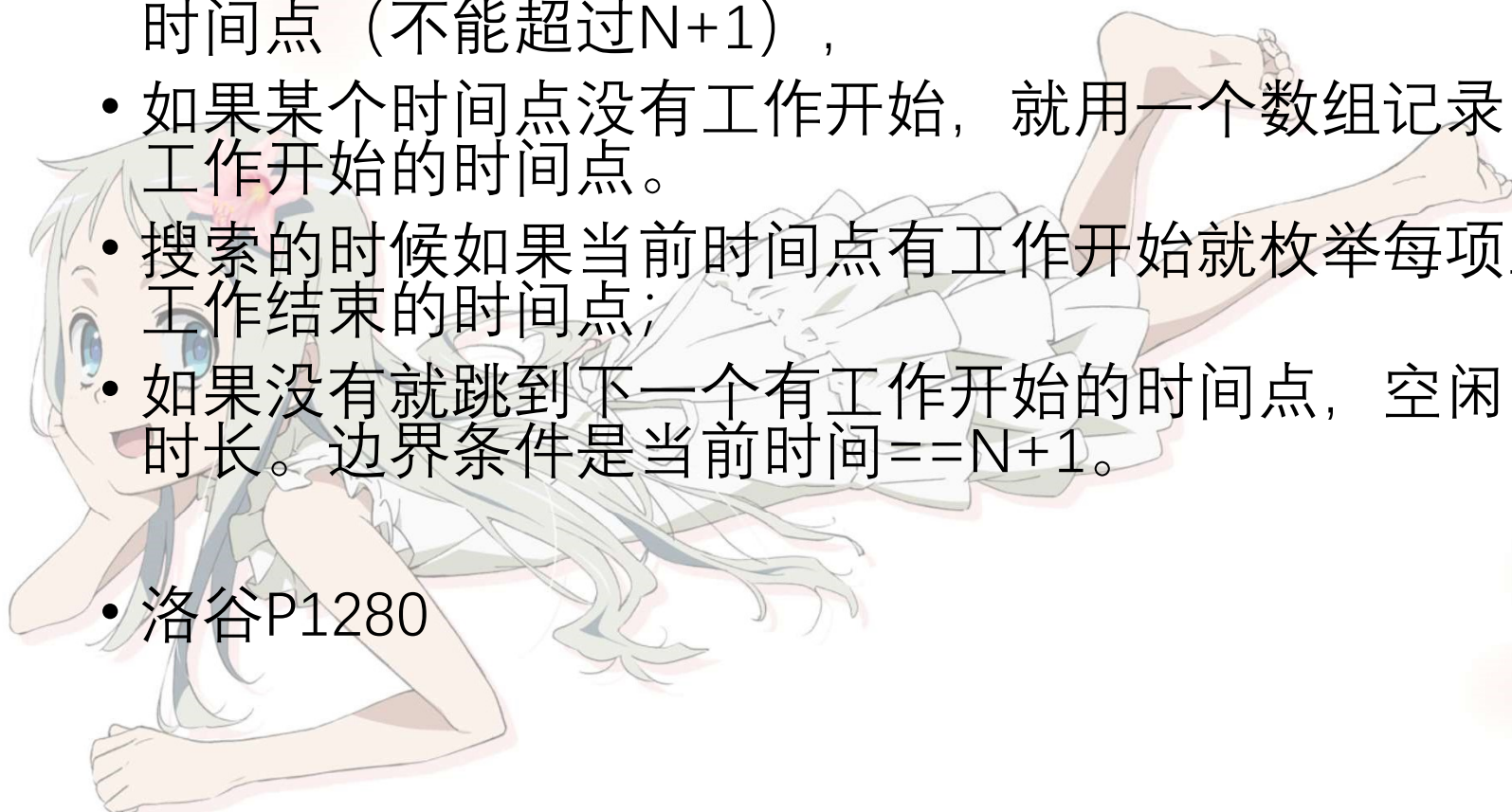
- Nike的一个工作日为N分钟，从第一分钟开始到第N分钟结束。当Nike到达单位后他就开始干活。如果在同一时刻有多个任务需要完成，Nike可以任选其中的一个来做，而其余的则由他的同事完成，反之如果只有一个任务，则该任务必需由Nike去完成，假如某些任务开始时刻Nike正在工作，则这些任务也由Nike的同事完成。如果某任务于第P分钟开始，持续时间为T分钟，则该任务将在第 $P+T-1$ 分钟结束。
- 写一个程序计算Nike应该如何选取任务，才能获得最大的空暇时间。



あの日見た
花の名前を
僕達はまだ
知らない。

例题7

- 记录某个在有工作开始的时间点做了从它开始的工作后可到达的时间点（不能超过 $N+1$ ），
- 如果某个时间点没有工作开始，就用一个数组记录它的下一个有工作开始的时间点。
- 搜索的时候如果当前时间点有工作开始就枚举每项工作并跳到该工作结束的时间点；
- 如果没有就跳到下一个有工作开始的时间点，空闲时间+=跳过的时长。边界条件是当前时间 $=N+1$ 。
- 洛谷P1280



あの日見た
花の名前を
僕達はまだ
知らない。

双向宽搜

- 双向BFS是一种对BFS的优化。对于那些已知起点状态和终点状态的BFS题目，都可以用双向BFS优化。
- 用两个队列分别存储从起点和终点开始搜索的状态，如果某一点同时被这两个队列搜索到，就停止搜索，总贡献是两者贡献之和。



あの日見た
花の名前を
僕達はまだ
知らない。

例题8

- 由于双向BFS的题目没什么新颖的，大家可以尝试一下用双向BFS写一下八数码问题。



あの日見た
花の名前を
僕達はまだ
知らない。

课后练习

- [Codevs 1018 单词接龙](#)
- [Codevs 1049 棋盘染色](#)
- [Codevs 1976 Queen数列](#)
- [Codevs 1099 字串变换](#)
- [Codevs 1569 最佳绿草](#)
- [洛谷P2534 \[AHOI2012\]铁盘整理](#)



あの日見た
花の名前を
僕達はまだ
知らない。

完结撒花

