NOIP 搜索 & 剪枝

quarter

2018 年 10 月 3 日

首先

~~迷之又被分到这个专题。~~

dfs 和 bfs 大家都会。所以全是题目。

复杂度科学一点的、不裸的题基本被讲烂了。所以很多那种你们想出了正解却不知道能不能过的题。

Description

有一个 1 至 2^n 的排列 $A\{1,...,2^n\}$。

可以执行的操作有 n 种，每种操作最多可以执行一次。第 i 种操作：将序列从左到右划分为 2^{n-i+1} 段，每段恰好包括 2^{i-1} 个数，然后整体交换其中两段。

求可以将数组 A 从小到大排序的不同的操作序列有多少个。

两个操作序列不同，当且仅当操作个数不同，或者至少一个操作不同（种类不同或者操作位置不同）。

$n \leq 12$

Solution

首先，任意一个合法的操作序列，我们可以改变其顺序，依然满足条件。那么这一类的操作序列的贡献，即为操作次数的阶乘。

Solution

首先，任意一个合法的操作序列，我们可以改变其顺序，依然满足条件。那么这一类的操作序列的贡献，即为操作次数的阶乘。

那么，现在只考虑种类编号递增的操作序列。第 i 种操作时，序列分成了大小为 2^{i-1} 的段，如果某个段不是递增且连续的，那么最后肯定不会满足条件。所以，在这种操作考虑完后，每个大小为 2^i 的段应当递增且连续。

Solution

首先，任意一个合法的操作序列，我们可以改变其顺序，依然满足条件。那么这一类的操作序列的贡献，即为操作次数的阶乘。

那么，现在只考虑种类编号递增的操作序列。第 i 种操作时，序列分成了大小为 2^{i-1} 的段，如果某个段不是递增且连续的，那么最后肯定不会满足条件。所以，在这种操作考虑完后，每个大小为 2^i 的段应当递增且连续。

当考虑第 i 种操作时：

- 如果不合条件的 2^i 大小的段超过 2 个，直接退出。
- 如果不存在，直接继续。
- 如果只有 1 个，交换其包含的 2^{i-1} 的两段，判断是否可行。
- 如果有 2 个，分别搜索 4 种交换方案。

Description

农夫约翰需要一些特定规格的木材 (共 n 块，长度不一定相同)，可是他只剩下一些大规格的木板 (共 m 块，长度不一定相同)。不过约翰可以将这些木板切割成他所需要的规格。

求约翰最多能够得到多少他所需要的木材。

$n \le 10^3$，$m \le 50$，长度 ≤ 32767。

Solution

显然可以二分答案。

判断答案为 k 是否可行，显然可以将木材排序，搜索最短的 k 块木材是否能得到。

Solution

显然可以二分答案。

判断答案为 k 是否可行，显然可以将木材排序，搜索最短的 k 块木材是否能得到。

然后剪枝：

- 从大到小搜索所需木材从哪块木板得到，因为长的木材限制较大。

Solution

显然可以二分答案。

判断答案为 k 是否可行，显然可以将木材排序，搜索最短的 k 块木材是否能得到。

然后剪枝：

- 从大到小搜索所需木材从哪块木板得到，因为长的木材限制较大。
- 如果两块所需木材长度相同，后搜索的那块的来源只需从前者来源开始枚举。

Solution

显然可以二分答案。

判断答案为 k 是否可行，显然可以将木材排序，搜索最短的 k 块木材是否能得到。

然后剪枝：

- 从大到小搜索所需木材从哪块木板得到，因为长的木材限制较大。
- 如果两块所需木材长度相同，后搜索的那块的来源只需从前者来源开始枚举。
- 当一块原料的长度比最短的需求还短，那么可以丢弃，如果丢弃总量 + 所需总量 > 原料总量，剪枝。

Description

给出一个数 S，输出所有约数和等于 S 的数。

$S \leq 2 * 10^9$，数据组数 ≤ 100。

Solution

$n = \prod_i p_i^{a_i} \quad \Rightarrow \quad \sigma(n) = \prod_i \sum_{j=0}^{a_i} p_i^j$

那么，我们可以通过枚举 p_i 及其 a_i 来搜索。

Solution

$n = \prod_i p_i^{a_i} \quad \Rightarrow \quad \sigma(n) = \prod_i \sum_{j=0}^{a_i} p_i^j$

那么，我们可以通过枚举 p_i 及其 a_i 来搜索。

- 若当前需要得到的 S 可以表示为一个未搜索过的质数与 1 的和，那么之前的数与这个质数的乘积是一个合法答案。

Solution

$n = \prod_i p_i^{a_i} \quad \Rightarrow \quad \sigma(n) = \prod_i \sum_{j=0}^{a_i} p_i^j$

那么，我们可以通过枚举 p_i 及其 a_i 来搜索。

- 若当前需要得到的 S 可以表示为一个未搜索过的质数与 1 的和，那么之前的数与这个质数的乘积是一个合法答案。
- 对于每一个使得 $(p+1)*(p+1) < S$ 的 p，枚举可能的 a_i，递归。

Description

统计 $[l, r]$ 内有多少个数，存在一个仅含有'6'、'8' 的数整除它。

$1 \le l \le r \le 10^{10}$

Solution

预处理出所有 $\leq r$ 的、仅含'6' 和'8' 的数 S，这些数里面存在一部分是其他这种数的倍数，去掉。

Solution

预处理出所有 $\leq r$ 的、仅含'6' 和'8' 的数 S，这些数里面存在一部分是其他这种数的倍数，去掉。

容斥：$ans_r = \sum_{T \subseteq S, T \neq \varnothing} (-1)^{|T|-1} * \lfloor \frac{r}{\text{lcm}(T)} \rfloor$

枚举每个 S_i 是否加入 T，一旦 $\lfloor \frac{r}{\text{lcm}(T)} \rfloor = 0$，剪枝。

Description

将一个正整数 i 的约数个数记为 $d(i)$。

如果对于一个正整数 k，$\forall 0 < i < k,\ d(k) > d(i)$，则 k 被称为反质数。

现在给定一个 n，求 n 以内最大的反质数。

$n \le 10^{100}$

Solution

先写一个高精度。

这道题就是求 $[1, n]$ 内，$d(x)$ 最大的 x，多个答案则选择最小的 x。

Solution

先写一个高精度。

这道题就是求 $[1,n]$ 内，$d(x)$ 最大的 x，多个答案则选择最小的 x。

$x=\prod_i p_i^{a_i}$，$d(x)=\prod_i a_i+1$，我们可以通过搜索 a_i 来寻找答案。

Solution

先写一个高精度。

这道题就是求 $[1, n]$ 内，$d(x)$ 最大的 x，多个答案则选择最小的 x。

$x = \prod_i p_i^{a_i}$，$d(x) = \prod_i a_i + 1$，我们可以通过搜索 a_i 来寻找答案。

- $\{a_i\}$ 一定不增。

Solution

先写一个高精度。

这道题就是求 $[1, n]$ 内，$d(x)$ 最大的 x，多个答案则选择最小的 x。

$x = \prod_i p_i^{a_i}$，$d(x) = \prod_i a_i + 1$，我们可以通过搜索 a_i 来寻找答案。

- $\{a_i\}$ 一定不增。
- 让 $a_i(i > 1)$ 增加 1，$d(x)$ 将乘上 $\frac{a_i+2}{a_i+1}$，而让 a_1 增加 $t = \lfloor \log_2 p_i \rfloor$，$d(x)$ 将乘上 $\frac{a_1+t+1}{a_1+1}$。如果 $\frac{a_i+2}{a_i+1} \leq \frac{a_1+t+1}{a_1+1}$，那么 a_i 必定不会加 1。

Solution

先写一个高精度。

这道题就是求 $[1, n]$ 内，$d(x)$ 最大的 x，多个答案则选择最小的 x。

$x = \prod_i p_i^{a_i}$ ，$d(x) = \prod_i a_i + 1$，我们可以通过搜索 a_i 来寻找答案。

- $\{a_i\}$ 一定不增。
- 让 $a_i(i > 1)$ 增加 1，$d(x)$ 将乘上 $\frac{a_i+2}{a_i+1}$，而让 a_1 增加 $t = \lfloor \log_2 p_i \rfloor$，$d(x)$ 将乘上 $\frac{a_1+t+1}{a_1+1}$。如果 $\frac{a_i+2}{a_i+1} \le \frac{a_1+t+1}{a_1+1}$，那么 a_i 必定不会加 1。
- 考虑用一个必定使得 $a_m = 0$ 的 p_m 来限制 a_i，设 k 是满足 $p_i^k > p_m$ 的最小整数，那么 $p_i^{a_i} > p_i^{a_i-k} * p_m$。若 a_i 存在于最优情况，那么 $\frac{a_i+1}{a_i-k+1} > \frac{a_m+2}{a_m+1} = 2 \Rightarrow a_i < 2*k-1$。$p_m$ 取使得 $\prod_{i=1}^{m} p_i > n$ 的最小 m 即可。

Description

一个 $n*m$ 的网格，k 种颜色，部分格子已经涂了某种颜色，现在需要将其他格子也涂上颜色，使得从 $(1,1)$ 到 (n,m) 的每条路径 (每次向下或向右走一格) 都不会出现重复颜色。求方案数，对 10^9+7 取模。

$n,m \le 10^3$，$k \le 10$

Solution

颜色数应大于等于步数，$n+m-1 \le k \Leftrightarrow n+m \le k+1 \le 11$，否则 `puts("0")`。

Solution

颜色数应大于等于步数，$n+m-1 \leq k \Leftrightarrow n+m \leq k+1 \leq 11$，否则 `puts("0")`。

然后搜每个位置的颜色，可以状压到每个位置的已经过的颜色。

可行性剪枝：未经过的颜色数小于剩余步数，剪掉。

Solution

颜色数应大于等于步数，$n+m-1 \le k \Leftrightarrow n+m \le k+1 \le 11$，否则 `puts("0")`。

然后搜每个位置的颜色，可以状压到每个位置的已经过的颜色。

可行性剪枝：未经过的颜色数小于剩余步数，剪掉。

对称性剪枝：如果涂上一个未出现过的颜色，涂哪一个都是等价的，那么只需搜其中一个。

Description

有 52 张牌排成一排 $\{0, 1, ..., 51\}$，然后可以把他们分成两半，然后交叉着洗牌 (一个确定的置换)。

洗完牌后，可能会出现一个错误，将一对相邻的牌调换了位置，但是每次洗牌最多只会犯一个错误。

现在给出牌最终的顺序，求最少洗了多少次 (保证不超过 10)，最少犯几次错误。

Solution

一个性质：确定了洗牌次数后，若真实的最终排列与不犯错误的最终排列，有 k 个位置不同，那么犯错次数 $\geq \lceil \frac{k}{2} \rceil$。

Solution

一个性质：确定了洗牌次数后，若真实的最终排列与不犯错误的最终排列，有 k 个位置不同，那么犯错次数 $\geq \lceil \frac{k}{2} \rceil$。

拿这个当剪枝，枚举洗牌次数和犯错次数，跑 idA*。

Description

你记录了 $[0,59]$ 这个时间段内到站的所有公交车 (数量 ≤ 300) 的时间，每辆车属于一条线路。

- 同一路线路的车的到站时间间隔相同。
- 每条线路在 $[0,59]$ 至少到达两辆车。
- 最多有 17 条线路。

求最少可能有多少条线路。

Solution

一条线路，可以通过第一辆、第二辆车来确定。

Solution

一条线路，可以通过第一辆、第二辆车来确定。

我们可以按照到站顺序来确定每辆车的归属情况。

- 如果它是线路第一辆车，那么先把他做一个标记。(同时，它的时间应在 $[0, 29]$)
- 如果是第二辆，枚举标记过的车辆来确定一条线路，如果可行，删除该线路的所有车。

Solution

一条线路，可以通过第一辆、第二辆车来确定。

我们可以按照到站顺序来确定每辆车的归属情况。

- 如果它是线路第一辆车，那么先把他做一个标记。(同时，它的时间应在 $[0, 29]$)
- 如果是第二辆，枚举标记过的车辆来确定一条线路，如果可行，删除该线路的所有车。

这样，随着搜索的深度增加，可选车辆快速减少，搜索规模大量降低。

Solution

一条线路，可以通过第一辆、第二辆车来确定。

我们可以按照到站顺序来确定每辆车的归属情况。

- 如果它是线路第一辆车，那么先把他做一个标记。(同时，它的时间应在 $[0, 29]$)
- 如果是第二辆，枚举标记过的车辆来确定一条线路，如果可行，删除该线路的所有车。

这样，随着搜索的深度增加，可选车辆快速减少，搜索规模大量降低。

同时可以加入若干合法性/最优性剪枝。比如：作为第一辆的车多于未确定的车。

Description

给定 n 个字符集为 {a,b,c,...,m} 的字符串 S_i，求一种 {a,b,c,...,m} 到 {0,1,2,3,4,5,6,7,8,9,+,*,=} 的映射 f，使得所有 $f(S_i)$ 均是表达式合法、且成立的等式。

$n \le 10^3$，$5 \le |S_i| \le 11$

Solution

先对约束条件较高的 +,*,= 进行搜索。

Solution

先对约束条件较高的 +,*,= 进行搜索。

然后，对于每一个等式，我们得到了每个数的位数，计算等号两侧的值域，无交集就可以剪枝。随着对应关系的确定，值域也会越来越小。

Solution

先对约束条件较高的 +,*,= 进行搜索。

然后，对于每一个等式，我们得到了每个数的位数，计算等号两侧的值域，无交集就可以剪枝。随着对应关系的确定，值域也会越来越小。

可以选择从低位开始搜，确定了低位的值，不相等可以剪枝。

Description

一个 $n*n$ 的网格，共 $2*n*(n+1)$ 条边，现在已经删除了一些边，问至少还需删去多少边，可以使得剩下的边不能构成正方形。

$n \leq 5$

Solution

数据范围较小，边的状态可以状压。

Solution

数据范围较小，边的状态可以状压。

- 每次考虑一个未破坏的正方形，枚举一条边破坏它。可以先考虑较小的正方形，因为这样更容易提前破坏掉大正方形。估价函数：两个有公共边的正方形连边，则每个连通分量至少删一条边。

Solution

数据范围较小，边的状态可以状压。

- 每次考虑一个未破坏的正方形，枚举一条边破坏它。可以先考虑较小的正方形，因为这样更容易提前破坏掉大正方形。估价函数：两个有公共边的正方形连边，则每个连通分量至少删一条边。
- 每次删掉一个至少能破坏一个正方形的边。应该先搜索能破坏正方形最多的边，计算出每条边可以破坏掉 a_i 个正方形，从大到小排序：$\max\{a_i\} = 1$ 时可以直接计算剩余步数；估价函数，最小的 x 使得 $\sum_{i=1}^{x} a_i \geq$ 剩余正方形数。

Solution

数据范围较小，边的状态可以状压。

- 每次考虑一个未破坏的正方形，枚举一条边破坏它。可以先考虑较小的正方形，因为这样更容易提前破坏掉大正方形。估价函数：两个有公共边的正方形连边，则每个连通分量至少删一条边。
- 每次删掉一个至少能破坏一个正方形的边。应该先搜索能破坏正方形最多的边，计算出每条边可以破坏掉 a_i 个正方形，从大到小排序：$\max\{a_i\} = 1$ 时可以直接计算剩余步数；估价函数，最小的 x 使得 $\sum_{i=1}^{x} a_i \geq$ 剩余正方形数。
- 这是一个可重覆盖问题，dlx

Description

一块 $h*w$ 的区域，存在障碍、空地、n 个建筑，从一个建筑到另一个建筑的花费为：路径上最长的连续空地的长度。

q 次询问：从建筑 s_i 到 t_i 的最小花费。

$h,w \le 2*10^3$，$n,q \le 2*10^5$

Solution

如果搞出了最小生成树，那么就只需在 kruskal 重构树上求 lca 就行了。然而边数达到 $O(n^2)$，暴力求边需要 $O(n * h * w)$。

Solution

如果搞出了最小生成树，那么就只需在 kruskal 重构树上求 lca 就行了。然而边数达到 $O(n^2)$，暴力求边需要 $O(n*h*w)$。

把所有建筑一起作为源点，跑 bfs，可以得到离每个位置最近的建筑及距离。然后，如果两个相邻位置的最近建筑不同，那么就将这对建筑连边。边数 $O(h*w)$。

Description

n 个点的有向图，边权 $\in \{1,2,3,4\}$，m 次修改边权/加边/删边，q 次询问：以 s_i 为起点，输出它到其他点的最短路。

$n \leq 5*10^2$，$m \leq 5*10^4$，$q \leq 5*10^3$

Solution

发现边权很小，直接 bitset 暴力 bfs。