NOIP2018模拟试题day1题解

2018.10.20

Cake

“这是一道看起来很原很原的题目

我也不知道它是不是原题

假装是自己原创的”

Cake

一点想法：

subtask1：只要不是满的O(N!*N)就行了

subtask2：状压DP，$O(2^N*N)$

subtask3&4：？？？

Cake

将所有的蛋糕按Ai排序，显然问题变为：

给定一个1~N的排列(Bi)，将其划分为尽可能少的上升子序列

Cake

将所有的蛋糕按Ai排序，显然问题变为：

给定一个1~N的排列(Bi)，将其划分为尽可能少的上升子序列

首先我们知道，答案的大小不会小于Bi的最长下降子序列的长度

因为这个子序列里的任意两个元素都不能出现在一组里

Cake

我们已经得到了答案的一个下界(记为W)

那么只要构造一个这样的合法解

就能同时证明这是最优解

Cake

考虑一个求最长下降子序列的二分算法

这个算法实质上是DP

每次通过二分找到唯一合法的转移

Cake

考虑一个求最长下降子序列的二分算法

这个算法实质上是DP

每次通过二分找到唯一合法的转移

我们可以发现以下的事实：

这个过程恰有W个状态

每个数字恰好会更新一个状态

更新每个状态的数字是单调增的

Cake

那么，在求最长下降子序列长度的过程中

我们把状态数组的每个位置改为栈

每次更新就是将一个新元素压入栈中

最后输出每个栈里的元素就可以了

具体实现可以用~~链表或者vector~~数组就好了

Cake

这是一道普及组级别的题目

考察了最长单调子序列的求法

只用到了二分法来解决问题

是不是一道标准的送分题呢

Bread

“本来K是10^5的”

Bread

一点想法：

subtask1：暴力枚举，O(N^6)

subtask2：前缀和优化，O(N^4)

subtask3：？？？

Bread

首先我们要知道方差的性质：

方差=平方的期望-期望的平方

期望很好计算，只需要知道每个特殊点在多少个矩形里出现过

那么平方的期望怎么算呢

Bread

对于一种选取方案

包含的特殊点个数的平方就相当于

包含了多少对特殊点

Bread

对于一种选取方案

包含的特殊点个数的平方就相当于

包含了多少对特殊点

我们只需要枚举一对特殊点

计算其被多少个矩形包含就行了

具体来说就是分别得出四个边界可以取几个值，再相乘

Bread

所以我们就愉快地做完了这道题啦

Bread

所以我们就愉快地做完了这道题啦

诶。。好像总方案数并不容易计算啊

Bread

这是一个经典问题，考虑枚举矩形下边界

由于受到上方特殊点的限制

合法的矩形必然包含于一个折现围成的区域

Bread

这是一个经典问题，考虑枚举矩形下边界

由于受到上方特殊点的限制

合法的矩形必然包含于一个折现围成的区域

可以用笛卡尔树来方便地计算贡献，构建时需要单调栈

但是坐标范围很大，注意离散化的处理：

1. 空行的处理

2. 上方没有特殊点的列的处理

Chocolate

“数据该怎么造呢

怎么造呢。。。？”

Chocolate

一点想法：

subtask1：哈希，排序，O(NlogN)

subtask2：一对串计入答案

当且仅当除去#后

一个串是另一个串的后缀

trie+树形DP

subtask3：。。。。。。

Chocolate

有了subtask2的提示，可以发现

一对串被计入答案的必要条件是：

- 取出第一个#前面的部分，其中一个是另一个的前缀
- 取出最后一个#后面的部分，其中一个是另一个的后缀

Chocolate

有了subtask2的提示，可以发现

一对串被计入答案的必要条件是：

- 取出第一个#前面的部分，其中一个是另一个的前缀
- 取出最后一个#后面的部分，其中一个是另一个的后缀

其实这也是充分条件

可以通过递归构造证明

Chocolate

那么类似subtask2

可以在两棵trie树上进行类似的统计

只要使用O(N^2)的暴力，就可以通过subtask3

Chocolate

考虑到这是一个离线的问题

可以在对第一棵树进行dfs的过程中

维护第一棵树上当前点到根的路径上的所有点

在第二棵树上的对应点

由于要支持对子树求和、对到根的一条链求和两种查询

只需要用两个树状数组配合dfs序来支持这些操作

时间复杂度O(LlogL)，其中L为字符串总长

Chocolate

这是一道基础数据结构题

考察了trie和树状数组的应用

完全在NOIP的考察范围内(理直气壮

Thanks！