

复杂度分析及应试技巧

黎伟诺

7.18.2024

补充一些比较重要的复杂度分析题目

均摊分析

均摊分析 (Amortized Analysis) 用来评价某个数据结构的一系列操作的平均代价，有时可能某个操作的代价特别高，但总体上来看也并非那么糟糕，可以形象的理解为把高代价的操作“分摊”到其他操作上去了，要求的就是均摊后的平均代价。

分析方法

累计分析：所有操作的时间和除以操作次数。

势能分析：每一个实际代价 c_i 都将数据结构从 D_{i-1} 改变为 D_i

定义均摊代价为： $c_i = c_i + \phi(D_i) - \phi(D_{i-1})$

ϕ 代表势能函数。

如果均摊代价要比实际代价大，那么我们就说势能增加，否则势能降低。

将所有操作的均摊代价相加可以得到均摊代价的和 = 实际代价和

$+ \phi(D_n) - \phi(D_0)$

假设初始 $\phi(D_0)$ 为 0。如果我们能证明 $\phi(D_n)$ 非负，那么我们就可以说明我们的均摊代价一定是实际代价的一个上界。

栈的 `push_back()`，连续 `pop_back()` k 次

C++ 的 `vector` 具体实现。

一个初始值为 0 的 k 位二进制计数器，每次加一。每次加一的运算次数为修改的位数。时间复杂度？

对应的势能函数：

- 1、栈内的元素个数
- 2、 $2 \times len - capacity$, (这个有点抽象，一般分析这个都是用累计分析)
- 3、计数器的 1 的个数

区间染色

有一个长度为 n 的颜色数组，需要支持区间染色，询问单点颜色以及所处连通块大小
强制在线

区间染色

用 set 维护颜色段，怎么分析复杂度？

复杂度小 trick

下面来看一些其它杂七杂八的复杂度分析。

打扑克

1.1 statement

Dove 和 Cicada 是好朋友, 他们经常在一起打扑克来消遣时光, 但是他们打的扑克有不同的玩法。

最开始时, 牌桌上会 n 个牌堆, 每个牌堆有且仅有一张牌, 第 i 个牌堆里那个扑克的编号为 i , 任意两张牌仅有标号不同。游戏会进行 m 轮, 每轮 Dove 可以执行下列操作之一:

- $1\ x\ y$, 将编号为 x, y 的牌所在的牌堆合并, 如果此时 x, y 已在同一牌堆中, 那么不进行任何操作。
- $2\ c$, 询问有多少对牌堆的牌数之差不少于 c 。形式化的, 对于当前的 r 个牌堆中, 有多少对 $i, j (i < j)$, 满足 $|size_i - size_j| \geq c$, 其中 $size_i$ 表示第 i 个牌堆的牌数。

每次 Cicada 都不能很快的回答出 Dove 的询问, 为了不让 Cicada 难堪, Dove 想要写一个小程序来帮助 Cicada, 但是 Dove 还要把妹学高考, 所以这个任务就交给你啦!

1.2 Input

第一行两个空格隔开的整数 n, m 。

接下来 m 行, 每行为 $1\ x\ y$ 或者 $2\ c$, 具体含义如上文所示。

1.3 Output

对于每个询问, 输出一行一个整数表示答案。

$$1 \leq n \leq 10^5, 1 \leq m \leq 3 \times 10^5$$

一个关键的性质：假如 $\sum a_i = n$ ，那么 a_i 的不同种类个数最多只有 \sqrt{n}
知道了这个性质我们就可以用 vector 维护一个有序的序列
vector 中存放牌数的大小和有多少堆这样的牌
并查集维护根的 size，相当于移除 szx, szy ，然后添加 $szx + szy$

Nonsense Time

HDU6635

Problem Description

You are given a permutation p_1, p_2, \dots, p_n of size n . Initially, all elements in p are frozen. There will be n stages that these elements will become available one by one. On stage i , the element p_{k_i} will become available.

For each i , find the longest increasing subsequence among available elements after the first i stages.

Input

The first line of the input contains an integer $T (1 \leq T \leq 3)$, denoting the number of test cases.

In each test case, there is one integer $n (1 \leq n \leq 50000)$ in the first line, denoting the size of permutation.

In the second line, there are n distinct integers $p_1, p_2, \dots, p_n (1 \leq p_i \leq n)$, denoting the permutation.

In the third line, there are n distinct integers $k_1, k_2, \dots, k_n (1 \leq k_i \leq n)$, describing each stage.

It is guaranteed that p_1, p_2, \dots, p_n and k_1, k_2, \dots, k_n are generated randomly.

随机数列的 LIS 级别是 $O(\sqrt{n})$ 。

先把所有数点亮，求一遍 LIS，得出 $O(\sqrt{n})$ 个关键点，这些关键点熄灭之前 LIS 长度不变，熄灭时直接重构。

时间复杂度 $O(n\sqrt{n}\log n)$

关于随机数据的一些性质

随机序列 LIS 是 $O(\sqrt{n})$

随机 ± 1 序列前缀和最大绝对值是 $O(\sqrt{n})$

$\sum a_i = n$ 的背包问题二进制分组是 $O(n\sqrt{n})$

最小原根 $O(\sqrt[4]{n})$

随机序列前缀最大值个数是 $O(\log n)$

随机父亲树高与最大度数都是 $O(\log n)$

约数表

$n \leq$	10^1	10^2	10^3	10^4	10^5	10^6	10^7	10^8	10^9
$\max\{\omega(n)\}$	2	3	4	5	6	7	8	8	9
$\max\{d(n)\}$	4	12	32	64	128	240	448	768	1344
$n \leq$	10^{10}	10^{11}	10^{12}	10^{13}	10^{14}	10^{15}	10^{16}	10^{17}	10^{18}
$\max\{\omega(n)\}$	10	10	11	12	12	13	13	14	15
$\max\{d(n)\}$	2304	4032	6720	10752	17280	26880	41472	64512	103680



应试技巧

时间安排

怎么规划对时间的安排？4 个小时面对 4 道题目

起码先 15 ~ 25 时间把题目都读一遍，建立对整套题的大概感受（哪些处于自己擅长的方向，哪些毫无想法）

一般来说 NOIP 的第一题就真的是第一题，第二第三题可能会根据个人情况有所变动。

想要拿一等，做出两道加上剩下的部分分是比较稳的

一道题目花了过长的时间想，或者预计写代码时间很长，应该稍微考虑一下做法麻烦了或者出问题了

最后一定要留足半个小时以上对拍、检查 freopen、检查文件夹之类的工作

打表

写一个很暴力很暴力的程序对小的 n 输出对应的结果
可以用来和之后的正解对拍，也可以用来寻找题目的规律或者思路

小凯的疑惑

P3951

NOIP2017 提高组 D1T1

题目描述

小凯手中有两种面值的金币，两种面值均为正整数且彼此互素。每种金币小凯都有无数个。在不找零的情况下，仅凭这两种金币，有些物品他是无法准确支付的。现在小凯想知道在无法准确支付的物品中，最贵的价值是多少金币？

注意：输入数据保证存在小凯无法准确支付的商品。

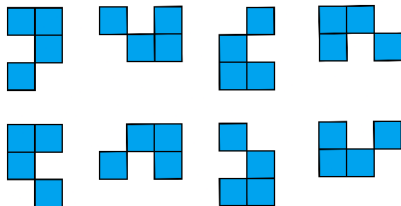
输入格式

两个正整数 a 和 b ，它们之间用一个空格隔开，表示小凯中金币的面值。

谜题：问号

2023 ICPC 南京

在一个 $n \times n$ 的网格内不重叠地放入尽可能多的问号拼图（QM 拼图），QM 拼图可以旋转和翻面，输出最多能放置的 QM 拼图数量以及对应的方案。



数列打表

让你对一个 n 求某种数列 f_n 的答案。
比如说长度为 $2n$ 的括号序列个数
找规律

数列打表

实在找不出规律了怎么办？

假如你的爆搜只能跑 n^2 ，但是 n 的范围是 10^5

你可以把你的程序挂着，输出 $n = 1, 2, \dots, 10^5$ 的值，然后开一个新的代码，定义数组为 $\text{int ans}[] = \{x_1, x_2, \dots\}$ ，这个就是你要交的程序

数列打表

表有可能过大，但是假如你知道 $f(n)$ 能比较快地推出 $f(n+1)$

分块打表

示例：求 $n! \pmod{10^9 + 7}$

暴力与部分分

对于题目内的不同范围试图去想对应复杂度的做法。

有些特定的数据范围，比如说保证该数据点的图是一棵树、保证 a_i 的种类数不会超过 5

这种特定的数据范围有可能提示着题目的最终解法。

也可以针对这些特殊范围编写相应的程序去得到部分分。

写暴力的一个分支，假如你的程序时间限制快用完了，直接输出无解然后退出。

示例：本次 ST02 测试 03 的 T1 字符串。

目前 CSP-S 以及 NOIP 系列比赛，均使用 Linux 作为操作系统
想要对拍，需要先熟悉 linux 的命令行基本命令

`cd`、`ls`、`pwd`

`bash pai.sh`

`g++ A.cpp -o A -std=c++11 -Wall -Wextra`

如何写对拍脚本

```
$ 1.bash ×
test > $ 1.bash
1  while true;do
2  ./gen > A.in
3  ./A < A.in > A.out
4  ./Astd < A.in > A.ans
5  if diff A.out A.ans -Z;then
6  |   printf "AC\n"
7  else
8  |   printf "WA\n"
9  |   break
10 fi
11 done
12 |
```

如何写 gen

选用好的随机数生成器，stop using rand()

```
gen.cpp x
test > gen.cpp > main()
1  #include <bits/stdc++.h>
2  using namespace std;
3  int main()
4  {
5      mt19937 rng((unsigned long long)(new char));
6      cout << rng()%10+1 << " " << rng()%10+1 << "\n";
7      return 0;
8  }
```

如何写 checker (部分题目限定)

某些题目需要输出方案，需要你写对应的 checker 检查方案是否合法
又或者答案输出为小数，需要比较精度

比赛最后的 check

末日 n 问？

对拍拍上了吗？

文件夹、freopen 对了吗

maxn 是否从调试时的 50 改回 200000 了？

多组数据成功清空了吗？

谢谢