

POJ 1200 【基础】

题目大意：

给出一个长度不超过 1600 万个字符的字符串，这个字符串中最多有 NC 个不同的字符出现 ($1 \leq NC \leq 255$)。问长度为 N 的不同子串有多少个。

输入：

第一行有两个空格隔开的整数 N 和 NC。第二行有一个给出的字符串。

输出：

一行，一个整数，即所求的子串个数。

题解：

Hash。将这些子串依次取出来求 Hash 值，然后在 Hash 表中标记，这样速度最快。我用的是网上说的 Rabin-Krap 写法，介绍见

<http://www.cnblogs.com/golove/p/3234673.html>。其实我的代码写得还有没有优化的地方，就是我每次都重新算了而没有利用前面的信息。

HDU 1880 【基础】

题目大意、输入输出要求见原题中文描述。

题解：

Hash 的模板题啊，虽然用 C++ 的 STL 里的 map 也能做，而且 Trie 肯定也是可以的，我还在网上看到有人暴力顺序搜索都能过……我用这题写了我的 BKDRHash 模板。关于各种字符串 Hash 的写法和对比，可以看

<http://blog.csdn.net/shifuwawa/article/details/5860946#>。

HDU 2072 【基础】

题目大意、输入输出要求见原题中文描述。

题解：

裸的 Hash 或者 Trie 字典树。我用的是 Hash，直接在原字符串上处理，所以比起模板有一点改动。

HDU 2648 【基础】

题目大意：

有 N 间商店 ($1 \leq N \leq 10000$)，每天都在涨价。现在给出每天这 N 间商店某一种货物的涨价额度，问每一天店名为 memory 的商店的价格是第几高的。

输入：

第一行有一个整数 N 。接下来有 N 行，每一行有一个字符串，表示一家商店的名字。然后有一行，包含一个整数 M ，表示有 M 天。下面有 M 组，每一组有 N 行，每一行先有一个整数 dp ，再有一个字符串 s ，两者用一个空格分隔，表示店名为 s 的商店涨价了 dp 元。

输出：

M 行，每一行有一个整数，第 i 行表示 memory 商店第 i 天的价格是第几高的。

题解：

Hash。其实可以直接套 map 的 STL，不过比较慢。排序后二分查找也可以，大概和 map 的速度一样。

POJ 3087 【基础】

题目大意：

已知两堆木片 $s1$ 和 $s2$ 的初始状态，其木片数均为 c ($1 \leq c \leq 100$)。先将 $s2$ 的最底下一块放到新的位置，然后将 $s1$ 的最底下一块叠在放出来的 $s2$ 的最底下的那块上面，依次类推操作，将它们相互交叉组合成一堆木片 $s12$ ，再将 $s12$ 的最底下的 c 块木片归为 $s1$ ，最顶的 c 块木片归为 $s2$ ，依此循环下去。问经过多少次新的组合之后， $s12$ 的状态和目标状态 $dest$ 相同，若永远不可能相同，则输出 “-1”。

输入：

第一行有一个整数 t ，表示一共有 t 组测试数据。

每一组测试数据第一行有一个整数 c ，下面三行依次是长度为 c 、 c 、 $2c$ 的字符串 $s1$ 、 $s2$ 和 $dest$ 。

输出：

每一组测试数据输出一行，包含两个整数（用空格分开），第一个整数是测试数据组编号，从 1 开始。第二个整数是要经过多少次组合才能达到要求。

题解：

模拟每一次操作，将得到的 s12 插入到 Hash 表中，然后查询即可。Hash 表所有的节点数我开了 1000 也够用了。

P0J 1840 【中等】

题目大意：

已知方程 $\sum (a_i \cdot x_i^3, i=0 \dots 4) = 0$ 中 a_i 和 x_i 的取值范围都是 $[-50, 50]$ ，给出 $a_0 \dots a_4$ ，求有多少个解。

输入：

一行，五个空格分隔的整数 $a_0 \ a_1 \ a_2 \ a_3 \ a_4$ 。

输出：

一行，一个整数，即解的个数。

题解：

直接枚举显然不行，一来是时间上会爆（ 100^5 ），二来需要高精度。因此将五个项分成两拨，一拨两个，一拨三个，先枚举一次算出其中一拨能取到的所有值，存入 Hash 表中，然后再枚举另外一拨能取到的所有值，取负后看看对应有多少个是存在的，加起来就得到答案。这题由于数据量大，不离散化而直接在数组里保存的话 int 会爆内存，改用 char（其实也有隐患，如果某个值可以被超过 127 种方式取得就会爆掉，但是最后测出来没有这种情况）。

和这题类似的是 HD0J 1496，不同的是方程改成了

$a \cdot x_0^2 + b \cdot x_1^2 + c \cdot x_2^2 + d \cdot x_3^2 = 0$ ， x_i 范围 $[-100, 100]$ 。做法基本一样，但是枚举的时候只需要枚举正半区就可以了，因为有一组解就肯定还有对应的 15 组解（ $2^4 - 1$ ），最后再 $\times 16$ 就可以了。

P0J 2785 【中等】

题目大意：

有 N 行（ $1 \leq N \leq 4000$ ），每一行有四个整数，这些整数的绝对值大小小于 2^{28} 。在每一列中取一个数字，问四个数字之和等于 0 的组合有多少。

输入：

第一行有一个整数 N ，接下来有 N 行，每一行有四个空格分隔的整数。

输出：

一行，包含一个整数，即符合要求的组合数。

题解：

和 POJ 1840 类似，也是分成两组，先求出其中一组的所有组合的和。但是这里我们不能直接使用桶排序一样来进行 Hash 标记，需要用到线性探测法：若 H_i 已经被占用了，则将 H_i 加上一个步进值，得到 $H(i+1)=(H_i+step)\%size$ ，直到找到一个没有被占用的位置。这里要求 $size>step$ 且两者互质。

POJ 2549 【中等】

题目大意：

给出 N 个 ($1\leq N\leq 1000$) 取值范围为 $[-536870912, +536870911]$ 的整数，求在里面选四个整数 $abcd$ 并满足 $a+b+c=d$ 的所有组合中， d 的最大值。

输入：

有若干组测试数据，每一组测试数据第一行有一个整数 N ， $N=0$ 时测试数据结束。接下来有 N 行，每一行有一个整数。

输出：

每一组测试数据输出一行，包含一个整数，即最大的 d 。

题解：

还是分成两部分， $a+b=d-c$ 。为了加快速度，需要对所有输入的整数进行排序，然后 d 从大到小搜索，只要找到一个符合的马上就退出。Hash 用的是加和取余，然后模拟链表。

POJ 3349 【中等】

题目大意：

有 N 片雪花 ($1\leq N\leq 100000$)，每一片雪花有六个臂，如果两片雪花旋转后六个臂对应的长度是一样的，则认为这两片雪花是相同的。求给出的雪花中有没有相同的。

输入：

第一行有一个整数 N 。接下来有 N 行，每一行有 6 个不大于 100000000 的正整数，表示一个臂长。给出的臂长顺序可能是顺时针也可能是逆时针，起始位置也可以是任何一个。

输出：

如果有相同的雪花，输出一行 “Twin snowflakes found.” 否则输出一行 “No two snowflakes are alike.”。

题解：

在 Discuss 里看到有很多种取 Hash 的方式，我用的是臂长求和对 14997 取余，并且偷懒用了静态表，也就是一次过给每一个 Hash 槽都分配了 100 个单元，有爆掉的可能性。实际上使用的内存不需要那么多，十分之一都不到。每一次读入的时候，就先检验对应的 Hash 槽中是否有相同的，如果没有再插入。检验的时候需要进行同构检验，我的做法是先找是否有相同的一个臂长，如果有则顺时针逆时针检测一次。