

2020牛客暑期多校训练营（第八场）

比赛情况

题号	A	B	C	D	E	F	G	H	I	J	K
状态	-	-	-	-	\emptyset	-	0	-	0	-	0

0 在比赛中通过 \emptyset 赛后通过! 尝试了但是失败了 - 没有尝试

比赛时间

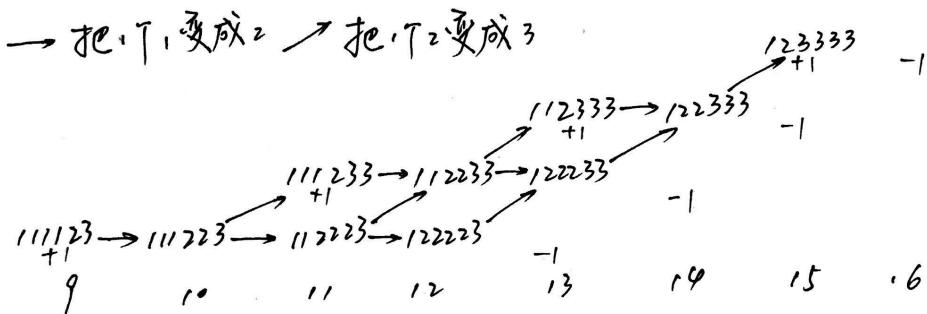
2020-08-03 12:00-17:00

题解

E - Enigmatic Partition

$f(x)$ 表示 x 由连续三个数字组成的划分有多少种，给出 l, r 求 $\sum_{k=l}^r f(k)$

显然只要能求得出来所有 $f(x)$ 就可以通过前缀和 $O(1)$ 得到 $\sum_{k=l}^r f(k)$



先放一个写题解必画的图...这是当起始数字 $i=1$ 长度固定为 $j=6$ 时的情况，差分一次我们发现 $+1$ 是从 $ij+3$ 开始往前跳两个数字一次， -1 是从 $(i+1)j+1$ 开始往前跳一个数字一次。于是再差分一次 ($+1$ 是跨度 2 的差分)，这样就可以枚举 i, j 后 $O(1)$ 计入贡献，最后做几次前缀和还原原数组即可。

```
#include<bits/stdc++.h>
#define ll long long
#define pii pair<int,int>
#define mp make_pair
#define fi first
#define se second
#define pb push_back
using namespace std;
const int N=1e5+10;
ll add[10*N], del[10*N], f[N];
```

```
int main()
{
    for(int i=1;i<N/3;i++)
        for(int j=3;j*i+3<N;j++)
    {
        add[i*j+3]++;
        add[(i+2)*j-1]--;
        del[(i+1)*j+1]++;
        del[(i+2)*j-1]--;
    }
    for(int i=2;i<N;i++)
        add[i] += add[i-2], del[i] += del[i-1], f[i] = add[i] - del[i] + f[i-1];
    for(int i=1;i<N;i++) f[i] += f[i-1];
    int t;
    scanf("%d",&t);
    for(int ca=1;ca<=t;ca++)
    {
        int l,r;
        scanf("%d%d",&l,&r);
        printf("Case #%d: %lld\n",ca,f[r]-f[l-1]);
    }
    return 0;
}
```

G - Game SET

每张卡牌有四个维度，每个维度有三种取向（或者一个通配符代表可以视作任一种），先要选出一组三张牌，使得每个维度要么都一样，要么都不一样。

先统计每种牌的数量，一共只有 \$81\$ 种牌，暴力枚举两种确定第三种就好（但其实是只要超过 \$21\$ 种必有可组的牌，复杂度更小）。

```
#include<bits/stdc++.h>
#define ll long long
#define pii pair<int,int>
#define mp make_pair
#define fi first
#define se second
#define pb push_back
using namespace std;
const int N=303;
int used[N],tim;
char s[4][50];
set<int>st[4][4][4][4],A,B,C,E,F,G;
vector<int>v[4];
int read()
{
    int x=0,f=1;char c=getchar();
    while(c<'0'||c>'9') {if(c=='-') f=-1;c=getchar();}
    if(f==1) x=c-'0';
    else x=c-'9'-1;
    if(x>3) x=3;
    if(x>6) x=6;
    if(x>9) x=9;
    if(x>12) x=12;
    if(x>15) x=15;
    if(x>18) x=18;
    if(x>21) x=21;
    if(x>24) x=24;
    if(x>27) x=27;
    if(x>30) x=30;
    if(x>33) x=33;
    if(x>36) x=36;
    if(x>39) x=39;
    if(x>42) x=42;
    if(x>45) x=45;
    if(x>48) x=48;
    if(x>51) x=51;
    if(x>54) x=54;
    if(x>57) x=57;
    if(x>60) x=60;
    if(x>63) x=63;
    if(x>66) x=66;
    if(x>69) x=69;
    if(x>72) x=72;
    if(x>75) x=75;
    if(x>78) x=78;
    if(x>81) x=81;
    if(x>84) x=84;
    if(x>87) x=87;
    if(x>90) x=90;
    if(x>93) x=93;
    if(x>96) x=96;
    if(x>99) x=99;
    if(x>102) x=102;
    if(x>105) x=105;
    if(x>108) x=108;
    if(x>111) x=111;
    if(x>114) x=114;
    if(x>117) x=117;
    if(x>120) x=120;
    if(x>123) x=123;
    if(x>126) x=126;
    if(x>129) x=129;
    if(x>132) x=132;
    if(x>135) x=135;
    if(x>138) x=138;
    if(x>141) x=141;
    if(x>144) x=144;
    if(x>147) x=147;
    if(x>150) x=150;
    if(x>153) x=153;
    if(x>156) x=156;
    if(x>159) x=159;
    if(x>162) x=162;
    if(x>165) x=165;
    if(x>168) x=168;
    if(x>171) x=171;
    if(x>174) x=174;
    if(x>177) x=177;
    if(x>180) x=180;
    if(x>183) x=183;
    if(x>186) x=186;
    if(x>189) x=189;
    if(x>192) x=192;
    if(x>195) x=195;
    if(x>198) x=198;
    if(x>201) x=201;
    if(x>204) x=204;
    if(x>207) x=207;
    if(x>210) x=210;
    if(x>213) x=213;
    if(x>216) x=216;
    if(x>219) x=219;
    if(x>222) x=222;
    if(x>225) x=225;
    if(x>228) x=228;
    if(x>231) x=231;
    if(x>234) x=234;
    if(x>237) x=237;
    if(x>240) x=240;
    if(x>243) x=243;
    if(x>246) x=246;
    if(x>249) x=249;
    if(x>252) x=252;
    if(x>255) x=255;
    if(x>258) x=258;
    if(x>261) x=261;
    if(x>264) x=264;
    if(x>267) x=267;
    if(x>270) x=270;
    if(x>273) x=273;
    if(x>276) x=276;
    if(x>279) x=279;
    if(x>282) x=282;
    if(x>285) x=285;
    if(x>288) x=288;
    if(x>291) x=291;
    if(x>294) x=294;
    if(x>297) x=297;
    if(x>300) x=300;
    if(x>303) x=303;
    if(x>306) x=306;
    if(x>309) x=309;
    if(x>312) x=312;
    if(x>315) x=315;
    if(x>318) x=318;
    if(x>321) x=321;
    if(x>324) x=324;
    if(x>327) x=327;
    if(x>330) x=330;
    if(x>333) x=333;
    if(x>336) x=336;
    if(x>339) x=339;
    if(x>342) x=342;
    if(x>345) x=345;
    if(x>348) x=348;
    if(x>351) x=351;
    if(x>354) x=354;
    if(x>357) x=357;
    if(x>360) x=360;
    if(x>363) x=363;
    if(x>366) x=366;
    if(x>369) x=369;
    if(x>372) x=372;
    if(x>375) x=375;
    if(x>378) x=378;
    if(x>381) x=381;
    if(x>384) x=384;
    if(x>387) x=387;
    if(x>390) x=390;
    if(x>393) x=393;
    if(x>396) x=396;
    if(x>399) x=399;
    if(x>402) x=402;
    if(x>405) x=405;
    if(x>408) x=408;
    if(x>411) x=411;
    if(x>414) x=414;
    if(x>417) x=417;
    if(x>420) x=420;
    if(x>423) x=423;
    if(x>426) x=426;
    if(x>429) x=429;
    if(x>432) x=432;
    if(x>435) x=435;
    if(x>438) x=438;
    if(x>441) x=441;
    if(x>444) x=444;
    if(x>447) x=447;
    if(x>450) x=450;
    if(x>453) x=453;
    if(x>456) x=456;
    if(x>459) x=459;
    if(x>462) x=462;
    if(x>465) x=465;
    if(x>468) x=468;
    if(x>471) x=471;
    if(x>474) x=474;
    if(x>477) x=477;
    if(x>480) x=480;
    if(x>483) x=483;
    if(x>486) x=486;
    if(x>489) x=489;
    if(x>492) x=492;
    if(x>495) x=495;
    if(x>498) x=498;
    if(x>501) x=501;
    if(x>504) x=504;
    if(x>507) x=507;
    if(x>510) x=510;
    if(x>513) x=513;
    if(x>516) x=516;
    if(x>519) x=519;
    if(x>522) x=522;
    if(x>525) x=525;
    if(x>528) x=528;
    if(x>531) x=531;
    if(x>534) x=534;
    if(x>537) x=537;
    if(x>540) x=540;
    if(x>543) x=543;
    if(x>546) x=546;
    if(x>549) x=549;
    if(x>552) x=552;
    if(x>555) x=555;
    if(x>558) x=558;
    if(x>561) x=561;
    if(x>564) x=564;
    if(x>567) x=567;
    if(x>570) x=570;
    if(x>573) x=573;
    if(x>576) x=576;
    if(x>579) x=579;
    if(x>582) x=582;
    if(x>585) x=585;
    if(x>588) x=588;
    if(x>591) x=591;
    if(x>594) x=594;
    if(x>597) x=597;
    if(x>600) x=600;
    if(x>603) x=603;
    if(x>606) x=606;
    if(x>609) x=609;
    if(x>612) x=612;
    if(x>615) x=615;
    if(x>618) x=618;
    if(x>621) x=621;
    if(x>624) x=624;
    if(x>627) x=627;
    if(x>630) x=630;
    if(x>633) x=633;
    if(x>636) x=636;
    if(x>639) x=639;
    if(x>642) x=642;
    if(x>645) x=645;
    if(x>648) x=648;
    if(x>651) x=651;
    if(x>654) x=654;
    if(x>657) x=657;
    if(x>660) x=660;
    if(x>663) x=663;
    if(x>666) x=666;
    if(x>669) x=669;
    if(x>672) x=672;
    if(x>675) x=675;
    if(x>678) x=678;
    if(x>681) x=681;
    if(x>684) x=684;
    if(x>687) x=687;
    if(x>690) x=690;
    if(x>693) x=693;
    if(x>696) x=696;
    if(x>699) x=699;
    if(x>702) x=702;
    if(x>705) x=705;
    if(x>708) x=708;
    if(x>711) x=711;
    if(x>714) x=714;
    if(x>717) x=717;
    if(x>720) x=720;
    if(x>723) x=723;
    if(x>726) x=726;
    if(x>729) x=729;
    if(x>732) x=732;
    if(x>735) x=735;
    if(x>738) x=738;
    if(x>741) x=741;
    if(x>744) x=744;
    if(x>747) x=747;
    if(x>750) x=750;
    if(x>753) x=753;
    if(x>756) x=756;
    if(x>759) x=759;
    if(x>762) x=762;
    if(x>765) x=765;
    if(x>768) x=768;
    if(x>771) x=771;
    if(x>774) x=774;
    if(x>777) x=777;
    if(x>780) x=780;
    if(x>783) x=783;
    if(x>786) x=786;
    if(x>789) x=789;
    if(x>792) x=792;
    if(x>795) x=795;
    if(x>798) x=798;
    if(x>801) x=801;
    if(x>804) x=804;
    if(x>807) x=807;
    if(x>810) x=810;
    if(x>813) x=813;
    if(x>816) x=816;
    if(x>819) x=819;
    if(x>822) x=822;
    if(x>825) x=825;
    if(x>828) x=828;
    if(x>831) x=831;
    if(x>834) x=834;
    if(x>837) x=837;
    if(x>840) x=840;
    if(x>843) x=843;
    if(x>846) x=846;
    if(x>849) x=849;
    if(x>852) x=852;
    if(x>855) x=855;
    if(x>858) x=858;
    if(x>861) x=861;
    if(x>864) x=864;
    if(x>867) x=867;
    if(x>870) x=870;
    if(x>873) x=873;
    if(x>876) x=876;
    if(x>879) x=879;
    if(x>882) x=882;
    if(x>885) x=885;
    if(x>888) x=888;
    if(x>891) x=891;
    if(x>894) x=894;
    if(x>897) x=897;
    if(x>900) x=900;
    if(x>903) x=903;
    if(x>906) x=906;
    if(x>909) x=909;
    if(x>912) x=912;
    if(x>915) x=915;
    if(x>918) x=918;
    if(x>921) x=921;
    if(x>924) x=924;
    if(x>927) x=927;
    if(x>930) x=930;
    if(x>933) x=933;
    if(x>936) x=936;
    if(x>939) x=939;
    if(x>942) x=942;
    if(x>945) x=945;
    if(x>948) x=948;
    if(x>951) x=951;
    if(x>954) x=954;
    if(x>957) x=957;
    if(x>960) x=960;
    if(x>963) x=963;
    if(x>966) x=966;
    if(x>969) x=969;
    if(x>972) x=972;
    if(x>975) x=975;
    if(x>978) x=978;
    if(x>981) x=981;
    if(x>984) x=984;
    if(x>987) x=987;
    if(x>990) x=990;
    if(x>993) x=993;
    if(x>996) x=996;
    if(x>999) x=999;
    if(x>1002) x=1002;
    if(x>1005) x=1005;
    if(x>1008) x=1008;
    if(x>1011) x=1011;
    if(x>1014) x=1014;
    if(x>1017) x=1017;
    if(x>1020) x=1020;
    if(x>1023) x=1023;
    if(x>1026) x=1026;
    if(x>1029) x=1029;
    if(x>1032) x=1032;
    if(x>1035) x=1035;
    if(x>1038) x=1038;
    if(x>1041) x=1041;
    if(x>1044) x=1044;
    if(x>1047) x=1047;
    if(x>1050) x=1050;
    if(x>1053) x=1053;
    if(x>1056) x=1056;
    if(x>1059) x=1059;
    if(x>1062) x=1062;
    if(x>1065) x=1065;
    if(x>1068) x=1068;
    if(x>1071) x=1071;
    if(x>1074) x=1074;
    if(x>1077) x=1077;
    if(x>1080) x=1080;
    if(x>1083) x=1083;
    if(x>1086) x=1086;
    if(x>1089) x=1089;
    if(x>1092) x=1092;
    if(x>1095) x=1095;
    if(x>1098) x=1098;
    if(x>1101) x=1101;
    if(x>1104) x=1104;
    if(x>1107) x=1107;
    if(x>1110) x=1110;
    if(x>1113) x=1113;
    if(x>1116) x=1116;
    if(x>1119) x=1119;
    if(x>1122) x=1122;
    if(x>1125) x=1125;
    if(x>1128) x=1128;
    if(x>1131) x=1131;
    if(x>1134) x=1134;
    if(x>1137) x=1137;
    if(x>1140) x=1140;
    if(x>1143) x=1143;
    if(x>1146) x=1146;
    if(x>1149) x=1149;
    if(x>1152) x=1152;
    if(x>1155) x=1155;
    if(x>1158) x=1158;
    if(x>1161) x=1161;
    if(x>1164) x=1164;
    if(x>1167) x=1167;
    if(x>1170) x=1170;
    if(x>1173) x=1173;
    if(x>1176) x=1176;
    if(x>1179) x=1179;
    if(x>1182) x=1182;
    if(x>1185) x=1185;
    if(x>1188) x=1188;
    if(x>1191) x=1191;
    if(x>1194) x=1194;
    if(x>1197) x=1197;
    if(x>1200) x=1200;
    if(x>1203) x=1203;
    if(x>1206) x=1206;
    if(x>1209) x=1209;
    if(x>1212) x=1212;
    if(x>1215) x=1215;
    if(x>1218) x=1218;
    if(x>1221) x=1221;
    if(x>1224) x=1224;
    if(x>1227) x=1227;
    if(x>1230) x=1230;
    if(x>1233) x=1233;
    if(x>1236) x=1236;
    if(x>1239) x=1239;
    if(x>1242) x=1242;
    if(x>1245) x=1245;
    if(x>1248) x=1248;
    if(x>1251) x=1251;
    if(x>1254) x=1254;
    if(x>1257) x=1257;
    if(x>1260) x=1260;
    if(x>1263) x=1263;
    if(x>1266) x=1266;
    if(x>1269) x=1269;
    if(x>1272) x=1272;
    if(x>1275) x=1275;
    if(x>1278) x=1278;
    if(x>1281) x=1281;
    if(x>1284) x=1284;
    if(x>1287) x=1287;
    if(x>1290) x=1290;
    if(x>1293) x=1293;
    if(x>1296) x=1296;
    if(x>1299) x=1299;
    if(x>1302) x=1302;
    if(x>1305) x=1305;
    if(x>1308) x=1308;
    if(x>1311) x=1311;
    if(x>1314) x=1314;
    if(x>1317) x=1317;
    if(x>1320) x=1320;
    if(x>1323) x=1323;
    if(x>1326) x=1326;
    if(x>1329) x=1329;
    if(x>1332) x=1332;
    if(x>1335) x=1335;
    if(x>1338) x=1338;
    if(x>1341) x=1341;
    if(x>1344) x=1344;
    if(x>1347) x=1347;
    if(x>1350) x=1350;
    if(x>1353) x=1353;
    if(x>1356) x=1356;
    if(x>1359) x=1359;
    if(x>1362) x=1362;
    if(x>1365) x=1365;
    if(x>1368) x=1368;
    if(x>1371) x=1371;
    if(x>1374) x=1374;
    if(x>1377) x=1377;
    if(x>1380) x=1380;
    if(x>1383) x=1383;
    if(x>1386) x=1386;
    if(x>1389) x=1389;
    if(x>1392) x=1392;
    if(x>1395) x=1395;
    if(x>1398) x=1398;
    if(x>1401) x=1401;
    if(x>1404) x=1404;
    if(x>1407) x=1407;
    if(x>1410) x=1410;
    if(x>1413) x=1413;
    if(x>1416) x=1416;
    if(x>1419) x=1419;
    if(x>1422) x=1422;
    if(x>1425) x=1425;
    if(x>1428) x=1428;
    if(x>1431) x=1431;
    if(x>1434) x=1434;
    if(x>1437) x=1437;
    if(x>1440) x=1440;
    if(x>1443) x=1443;
    if(x>1446) x=1446;
    if(x>1449) x=1449;
    if(x>1452) x=1452;
    if(x>1455) x=1455;
    if(x>1458) x=1458;
    if(x>1461) x=1461;
    if(x>1464) x=1464;
    if(x>1467) x=1467;
    if(x>1470) x=1470;
    if(x>1473) x=1473;
    if(x>1476) x=1476;
    if(x>1479) x=1479;
    if(x>1482) x=1482;
    if(x>1485) x=1485;
    if(x>1488) x=1488;
    if(x>1491) x=1491;
    if(x>1494) x=1494;
    if(x>1497) x=1497;
    if(x>1500) x=1500;
    if(x>1503) x=1503;
    if(x>1506) x=1506;
    if(x>1509) x=1509;
    if(x>1512) x=1512;
    if(x>1515) x=1515;
    if(x>1518) x=1518;
    if(x>1521) x=1521;
    if(x>1524) x=1524;
    if(x>1527) x=1527;
    if(x>1530) x=1530;
    if(x>1533) x=1533;
    if(x>1536) x=1536;
    if(x>1539) x=1539;
    if(x>1542) x=1542;
    if(x>1545) x=1545;
    if(x>1548) x=1548;
    if(x>1551) x=1551;
    if(x>1554) x=1554;
    if(x>1557) x=1557;
    if(x>1560) x=1560;
    if(x>1563) x=1563;
    if(x>1566) x=1566;
    if(x>1569) x=1569;
    if(x>1572) x=1572;
    if(x>1575) x=1575;
    if(x>1578) x=1578;
    if(x>1581) x=1581;
    if(x>1584) x=1584;
    if(x>1587) x=1587;
    if(x>1590) x=1590;
    if(x>1593) x=1593;
    if(x>1596) x=1596;
    if(x>1599) x=1599;
    if(x>1602) x=1602;
    if(x>1605) x=1605;
    if(x>1608) x=1608;
    if(x>1611) x=1611;
    if(x>1614) x=1614;
    if(x>1617) x=1617;
    if(x>1620) x=1620;
    if(x>1623) x=1623;
    if(x>1626) x=1626;
    if(x>1629) x=1629;
    if(x>1632) x=1632;
    if(x>1635) x=1635;
    if(x>1638) x=1638;
    if(x>1641) x=1641;
    if(x>1644) x=1644;
    if(x>1647) x=1647;
    if(x>1650) x=1650;
    if(x>1653) x=1653;
    if(x>1656) x=1656;
    if(x>1659) x=1659;
    if(x>1662) x=1662;
    if(x>1665) x=1665;
    if(x>1668) x=1668;
    if(x>1671) x=1671;
    if(x>1674) x=1674;
    if(x>1677) x=1677;
    if(x>1680) x=1680;
    if(x>1683) x=1683;
    if(x>1686) x=1686;
    if(x>1689) x=1689;
    if(x>1692) x=1692;
    if(x>1695) x=1695;
    if(x>1698) x=1698;
    if(x>1701) x=1701;
    if(x>1704) x=1704;
    if(x>1707) x=1707;
    if(x>1710) x=1710;
    if(x>1713) x=1713;
    if(x>1716) x=1716;
    if(x>1719) x=1719;
    if(x>1722) x=1722;
    if(x>1725) x=1725;
    if(x>1728) x=1728;
    if(x>1731) x=1731;
    if(x>1734) x=1734;
    if(x>1737) x=1737;
    if(x>1740) x=1740;
    if(x>1743) x=1743;
    if(x>1746) x=1746;
    if(x>1749) x=1749;
    if(x>1752) x=1752;
    if(x>1755) x=1755;
    if(x>1758) x=1758;
    if(x>1761) x=1761;
    if(x>1764) x=1764;
    if(x>1767) x=1767;
    if(x>1770) x=1770;
    if(x>1773) x=1773;
    if(x>1776) x=1776;
    if(x>1779) x=1779;
    if(x>1782) x=1782;
    if(x>1785) x=1785;
    if(x>1788) x=1788;
    if(x>1791) x=1791;
    if(x>1794) x=1794;
    if(x>1797) x=1797;
    if(x>1800) x=1800;
    if(x>1803) x=1803;
    if(x>1806) x=1806;
    if(x>1809) x=1809;
    if(x>1812) x=1812;
    if(x>1815) x=1815;
    if(x>1818) x=1818;
    if(x>1821) x=1821;
    if(x>1824) x=1824;
    if(x>1827) x=1827;
    if(x>1830) x=1830;
    if(x>1833) x=1833;
    if(x>1836) x=1836;
    if(x>1839) x=1839;
    if(x>1842) x=1842;
    if(x>1845) x=1845;
    if(x>1848) x=1848;
    if(x>1851) x=1851;
    if(x>1854) x=1854;
    if(x>1857) x=1857;
    if(x>1860) x=1860;
    if(x>1863) x=1863;
    if(x>1866) x=1866;
    if(x>1869) x=1869;
    if(x>1872) x=1872;
    if(x>1875) x=1875;
    if(x>1878) x=1878;
    if(x>1881) x=1881;
    if(x>1884) x=1884;
    if(x>1887) x=1887;
    if(x>1890) x=1890;
    if(x>1893) x=1893;
    if(x>1896) x=1896;
    if(x>1899) x=1899;
    if(x>1902) x=1902;
    if(x>1905) x=1905;
    if(x>1908) x=1908;
    if(x>1911) x=1911;
    if(x>1914) x=1914;
    if(x>1917) x=1917;
    if(x>1920) x=1920;
    if(x>1923) x=1923;
    if(x>1926) x=1926;
    if(x>1929) x=1929;
    if(x>1932) x=1932;
    if(x>1935) x=1935;
    if(x>1938) x=1938;
    if(x>1941) x=1941;
    if(x>1944) x=1944;
    if(x>1947) x=1947;
    if(x>1950) x=1950;
    if(x>1953) x=1953;
    if(x>1956) x=1956;
    if(x>1959) x=1959;
    if(x>1962) x=1962;
   
```

```
while(c>='0'&&c<='9') {x=x*10+c-'0';c=getchar();}
```

```
return x*f;
```

```
}
```

```
void getS(char *p)
```

```
{
```

```
    char c=getchar();int i=0;
```

```
    while(c!=' ')c=getchar();
```

```
    c=getchar();
```

```
    while(c!=']')p[i++]=c,c=getchar();
```

```
    p[i]='\0';
```

```
}
```

```
int main()
```

```
{
```

```
    int t=read();
```

```
    for(int ca=1;ca<=t;ca++)
```

```
    {
```

```
        memset(used,0,sizeof(used)),tim=0;
```

```
        for(int a=1;a<=3;a++)
```

```
            for(int b=1;b<=3;b++)
```

```
                for(int c=1;c<=3;c++)
```

```
                    for(int d=1;d<=3;d++)
```

```
                        set<int>().swap(st[a][b][c][d]);
```

```
        int n=read();
```

```
        for(int i=1;i<=n;i++)
```

```
        {
```

```
            getS(s[0]),getS(s[1]),getS(s[2]),getS(s[3]);
```

```
            vector<int>().swap(v[0]),vector<int>().swap(v[1]);
```

```
            vector<int>().swap(v[2]),vector<int>().swap(v[3]);
```

```
            if(strcmp(s[0],"*")==0)v[0].pb(1),v[0].pb(2),v[0].pb(3);
```

```
            else if(strcmp(s[0],"one")==0)v[0].pb(1);
```

```
            else if(strcmp(s[0],"two")==0)v[0].pb(2);
```

```
            else if(strcmp(s[0],"three")==0)v[0].pb(3);
```

```
            if(strcmp(s[1],"*")==0)v[1].pb(1),v[1].pb(2),v[1].pb(3);
```

```
            else if(strcmp(s[1],"diamond")==0)v[1].pb(1);
```

```
            else if(strcmp(s[1],"squiggle")==0)v[1].pb(2);
```

```
            else if(strcmp(s[1],"oval")==0)v[1].pb(3);
```

```
            if(strcmp(s[2],"*")==0)v[2].pb(1),v[2].pb(2),v[2].pb(3);
```

```
            else if(strcmp(s[2],"solid")==0)v[2].pb(1);
```

```
            else if(strcmp(s[2],"striped")==0)v[2].pb(2);
```

```
            else if(strcmp(s[2],"open")==0)v[2].pb(3);
```

```
            if(strcmp(s[3],"*")==0)v[3].pb(1),v[3].pb(2),v[3].pb(3);
```

```
            else if(strcmp(s[3],"red")==0)v[3].pb(1);
```

```
            else if(strcmp(s[3],"green")==0)v[3].pb(2);
```

```
            else if(strcmp(s[3],"purple")==0)v[3].pb(3);
```

```
        for(int a:v[0])for(int b:v[1])
        for(int c:v[2])for(int d:v[3])
            st[a][b][c][d].insert(i);
    }
    printf("Case #%d: ",ca);
    int f=0;
    for(int a=1;a<=3&&!f;a++)
    for(int b=1;b<=3&&!f;b++)
    for(int c=1;c<=3&&!f;c++)
    for(int d=1;d<=3&&!f;d++)
    {
        if(st[a][b][c][d].empty())continue;
        if(st[a][b][c][d].size()>=3)
        {
            int cnt=0;
            for(int x:st[a][b][c][d])
            {
                printf("%d ",x),cnt++;
                if(cnt==3)break;
            }
            puts(" "),f=1;
        }
        else
        {
            for(int p=1;p<=3&&!f;p++)
            for(int q=1;q<=3&&!f;q++)
            for(int g=1;g<=3&&!f;g++)
            for(int h=1;h<=3&&!f;h++)
            {
                if(st[p][q][g][h].empty())continue;
                int t1,t2,t3,t4;
                if(p==a)t1=a;else t1=6-a-p;
                if(b==q)t2=b;else t2=6-b-q;
                if(c==g)t3=c;else t3=6-c-g;
                if(d==h)t4=d;else t4=6-d-h;

                if(st[t1][t2][t3][t4].empty())continue;
                tim++;int x=0,y=0,z=0;
                for(int
k:st[a][b][c][d])if(used[k]!=tim){used[k]=tim,x=k;break;}
                    for(int
k:st[p][q][g][h])if(used[k]!=tim){used[k]=tim,y=k;break;}
                    for(int
k:st[t1][t2][t3][t4])if(used[k]!=tim){used[k]=tim,z=k;break;}
                        if(x&y&z)printf("%d %d %d\n",x,y,z),f=1;
                    }
                }
            if(!f)puts("-1");
        }
    }
```

```

    return 0;
}

```

I - Interesting Computer Game

每回合给出两个数字，你可以从中选一个，问最终最多能选得多少种数字。

把每回合的两个数字连边，每个连通块大小为 \$n\$ 则至少能选 \$n-1\$ 个（仅不选dfs树根节点），当仅有环可以选 \$n\$ 个。

```

#include<bits/stdc++.h>
#define ll long long
#define pii pair<int,int>
#define mp make_pair
#define fi first
#define se second
#define pb push_back
using namespace std;
const int N=1e5+10;
ll read()
{
    ll x=0,f=1;char c=getchar();
    while(c<'0'||c>'9'){if(c=='-')f=-1;c=getchar();}
    while(c>='0'&&c<='9'){x=x*10+c-'0';c=getchar();}
    return x*f;
}
int a[N],b[N],num[N*2],tot,father[N*2],sz[N*2],e[N*2];
int find(int x){return x==father[x]?x:father[x]=find(father[x]);}
int main()
{
    int t=read();
    for(int ca=1;ca<=t;ca++)
    {
        tot=0;
        int n=read();
        for(int i=1;i<=n;i++)a[i]=read(),b[i]=read(),num[++tot]=a[i],num[++tot]=b[i];
        sort(num+1,num+1+tot);
        tot=unique(num+1,num+1+tot)-num-1;
        for(int i=1;i<=tot;i++)father[i]=i,sz[i]=1,e[i]=0;
        for(int i=1;i<=n;i++)
        {
            a[i]=lower_bound(num+1,num+1+tot,a[i])-num;
            b[i]=lower_bound(num+1,num+1+tot,b[i])-num;
            int fu=find(a[i]),fv=find(b[i]);
            if(fu!=fv){father[fu]=fv,sz[fv]+=sz[fu],e[fv]+=e[fu];}
            e[fv]++;
        }
    }
}

```

```
        }
        int res=0;
        for(int i=1;i<=tot;i++)
            if(father[i]==i)res+=(e[i]>sz[i]-1)?sz[i]:sz[i]-1;
        printf("Case #%d: %d\n",ca,res);

    }
    return 0;
}
```

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:wangzai_milk:20200803%E6%AF%94%E8%B5%9B%E8%AE%B0%E5%BD%95&rev=1596629405

Last update: 2020/08/05 20:10

