A
遮体，胜赛选拔赛的时候，出过一场类似的题目了吧，那个时候问最多可以分多少组，每组有 M 个数，都不相同。
现在是每组两个数。
首先记录下每个数，假设现在枚举到 i，最优情况下肯定是 i 先不配对，让其他的数配对。
假设[1,2,..i - 1,i + 1...n]中数的个数为 tot,且出现次数最多的数为 max 的话
1)如果 max >= tot - max，则[1,2,..i - 1,i + 1...n]的最好配对方法是每个数都与 max 配对，剩下 max - (tot - max)个数没配对
2)如果 max < tot - max，则可以剩下 tot % 2 个数
看看剩下的数与 i 出现次数做比对，如果 app(i) > 剩下的数，ans += i;

B
用单调栈预处理出当以 a[i]为最小值的时候，最左向左延伸到哪里，最右向右延伸到哪里
不妨设为 lt[i],rt[i].
伪代码: a[0] = a[n + 1] = -1;
```
for(int i = 1; i <= n; i ++) {
    lt[i] = i;
    while(a[i] <= a[lt[i] - 1]) lt[i] = lt[lt[i] - 1];
}
```
rt[i]同样处理。当然，有很多种处理方法。
用 max_len[i]表示，当长度为 i 的时候，延伸长度不小于 i 的 ai 的最大值,
for(int i = 1; i <= n; i ++) max_len[rt[i] - lt[i] + 1] = max(max_len[rt[i] - lt[i] + 1],a[i]);
max_len[i] = max(max_len[i],max_len[i + 1])
对于第一问，处理一个后缀，dp[i] = max(dp[i + 1],max_len[i] * i)
输入一个 x，判掉不合法的之后，直接输出 dp[x]
对于第二问，和第一问类似，处理一个前缀，然后需要二分。

C
非常暴力就可以过了，可以进行一个优化
假设给出一个 N 的串 S，和一个长度为 M 的串 STR，如果 N 长度远大与 M，如果快速判断 STR 是 S 的子串
用 next[i(i = 'a'-'z')][pos]表示表示 i 这个字符，下一次出现位置>=pos 的最近位置是哪里
那么
```
for(int i = 1,pos = 0; i <= M; i ++) {
        int x = STR[i] - 'a';
        int new_pos = next[i][pos + 1];
        if(new_pos > N) return "can not find";
        else pos = new_pos;
    }
    return "find";
}
```
或者用一个数组 pos[26][N + 10]的数组存下每个字符出现的位置，二分查找下一个位置

D
枚举任意三个点，然后枚举有哪些点被这 3 个点覆盖了。C(N,3) * N

二分面积最大值，BFS 能否达(1 << N )- 1 这个状态(这里不需要 DP 了)

E
最初始的时候这题是没有取摸的，但是因为考虑到整体题目难度问题，将高精度去掉了，否则高精度加减乘除（低精）都有了吧。
数的形式是:
1
1 3
1 3 5 7
1 3 5 7 9 11 13 15
的形式。
要使序列是连续上升的一段，必然是从 1 开始的一段。
对于 1 3 5 7 9 11...这种，前 i 项的和是 i^2（这里，我偷了一下懒，将数出成 1 3 5 的情况，而不是 1，2，3）
枚举连续上升子序列的最后一项,那么和就是 sigma(i^2 - j^2)，i^2 - j ^ 2 表示子序列是第 j + 1 项到第 i 项的。
求和就是 sum(i) i^3 - i * (i - 1) * (2 * i - 1) / 6。
还剩下一步就是 ans = sigma(sum(i)) (i = 1..n)
自己推，不写出来了
有除以 6 和除以 4 的操作，hint 有提到,(a / b) % c = a * b' % c(c 为质数), b' 即 b 的逆元，即 b' * b % c = 1,因为摸数为 10007，所以 for 意下找 b'就行了。
mod 很大的时候，用扩展 gcd 或者 b' = b^(c - 2)，根据欧拉定理,b ^ (prime - 1) % prime = 1

```
for(int i = 0; n; i ++) {
    int m = min(n,1 << i);
    ans += cal(m) //cal 就是上面的东西。
}
return ans % mod;
```

F
怎么看都是觉得是全场最水的题目，直接判断数 N 是否两个不同质数相乘。

```
for(int i = 2; i * i <= n; i ++)
    if(n % i == 0) {
        if((n / i) % i == 0) return "NO";
        tot ++;
        while(n % i == 0) n /= i;
    }
if(n > 1) tot ++;
return tot == 2;
```

或者，2 个不同质数相乘，不就只有 4 个约数，1,p,q,pq 么。

```
int tot = 0;
for(int i = 1; i * i <= n; i ++)
    if(n % i == 0) {
        tot ++;
```

```
        if(i * i != n) tot ++;
    }
return tot == 4;
```