

# D Choice II——by 董舒印

## Description

---

在 DPY 的努力争取下，名额从 2 个变成了 3 个。好吧，Thor 又要重新计算了，囧。  
他给 Thor 一段长度为  $n$  的数列和一个数  $S$ ，表示他对  $n$  个同学的看好程度  $A[i]$ ，要求 Thor 从里面选择三个人，使得这三个人的被看好程度之和大于  $S$ 。  
现在，问 Thor 有多少的选择方式。

## Input

---

多组数据。

每组数据有 2 行。

第一行有 2 个整数  $n, s$  ( $1 \leq n \leq 1000, 0 \leq s \leq 200000$ )。

第二行有  $n$  个整数  $A[i]$  ( $0 \leq A[i] \leq 200000$ ) 表示每个人被看好程度。

## Output

---

对于每组数据输出一行一个整数  $Ans$  表示 Thor 的选择方案数。

## Sample Input

---

```
5 6
2 3 5 4 2
6 9
2 7 2 5 6 1
```

## Sample Output

---

```
10
14
```

## Hint

---

$1000^3$  是会超时的

# 解题分析

算法复杂度到  $O(100000000)$  基本就会跪了 ..... 所以暴力搜索行不通。

那怎么办呢????

其实就是两层暴力一层二分 - #

二分的算法复杂度是  $O(\log_2 n)$  (以 2 为底)

算法复杂度立刻降为原来的  $1/100!$

貌似有同学不清楚为什么是  $O(\log_2 n)$ ，简单解释一下就是，因为每次能处理  $1/2$  的数据，所以就是  $O(\log_2 n)$  ..... 可以自己简单试一下，很容易理解的。

二分这个查找方式很强，但是不同的题写起来不一样，很容易出错，所以建议大家多多练习。

## 参考代码

```
#include<iostream>
#include<cstdio>
using namespace std;
int a[200500], s, n; //定义成全局变量方便函数直接调用。
long long ans; //我估计了一下最大结果貌似会爆 int……但是亲测没有爆 =
说明数据不够丧病。
void solve(int i, int j)
{
    int l=j+1, r=n, m; //l 为左边界, r 为右边界
    while(l<=r) //注意这里是小于等于号, 想想为什么。或者动手模拟一下。
    {
        m=(l+r)/2;
        if(a[i]+a[j]+a[m]>s)
        {
            ans+=r-m+1; //这个是这个二分比较独特的地方。
            r=m-1;
        }
        else
            l=m+1;
    }
}
int main()
{
    while(scanf("%d%d", &n, &s) != EOF)
    {
        ans=0;
        for(int i=1; i<=n; i++)
            scanf("%d", &a[i]);
        for(int i=1; i<=n; i++)
            for(int j=i+1; j<=n; j++)
                if(a[i]>a[j])
                    swap(a[i], a[j]);
        //我个人的冒泡排序写法, swap 函数在 iostream 头文件里有。
        for(int i=1; i<=n; i++)
            for(int j=i+1; j<=n; j++)
                solve(i, j);
        printf("%lld\n", ans); //注意长整形的 C 风格输出要用 %lld, 这个是
Linux 下的……windows 下是 %I64d, 咱们的 OJ 是 Linux 下的, 如果去别的 OJ
做题要测评机是哪种 = 我表示曾经被这个坑了一下午。
    }
}
```