程序设计方法与艺术

# 解题报告

班　　级：计科 24—3 班、计科 24—4 班
组　　长：2024210861 贺雨萱
组　　员：2024210858 傅家琪
2024210908 曹蕊
2024213885 朱铭

# 题目 A 植被保护（2024213885 朱铭）

H 市现在正在大力发展工业，众所周知发展工业，会影响当地的植被 覆盖率。第一年 H 市的植被覆盖数为 N 平方千米，当植被覆盖数达到 M 平方 千米以下时，则说明当地已经严重污染，植被覆盖不足。假设 H 市的植被覆 盖数每年以 K%的减少。请问多少年之后 H 市将会严重污染？

输入说明：一行包含三个整数 N,M,K（ $31 \le N,M \le 2-1, 1 \le K < 100$ ），N 表 示第一年 H 市的植被覆盖数,M 表示 M 平方千米以下时，则说明当地已经严重 污染，K 表示 H 市的植被覆盖数每年以 K%的减少。

**解题思路：**

这道题目很明显是道循环题，关键在于优化。题目给出了三个参数：

- N：初始植被覆盖面积（平方千米）
- M：植被覆盖低于 M 时判定为严重污染
- K：每年植被覆盖减少的百分比

要求计算多少年后植被覆盖会低于 M。

**具体解法：**

方法一：逐年计算法

这种方法比较直观，适合本题的数据范围：

1. 从初始面积 N 开始计算
2. 每过一年，将当前面积乘以 (100-K)/100
3. 重复这个过程，直到面积小于 M
4. 统计经过的年数就是答案

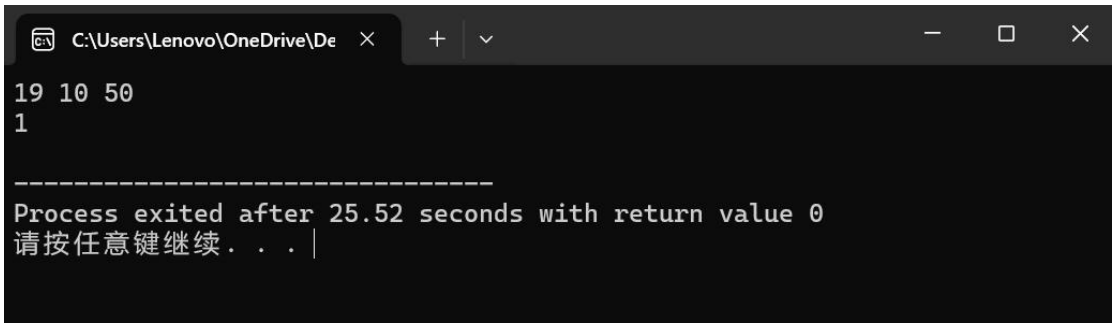
方法二：数学公式法

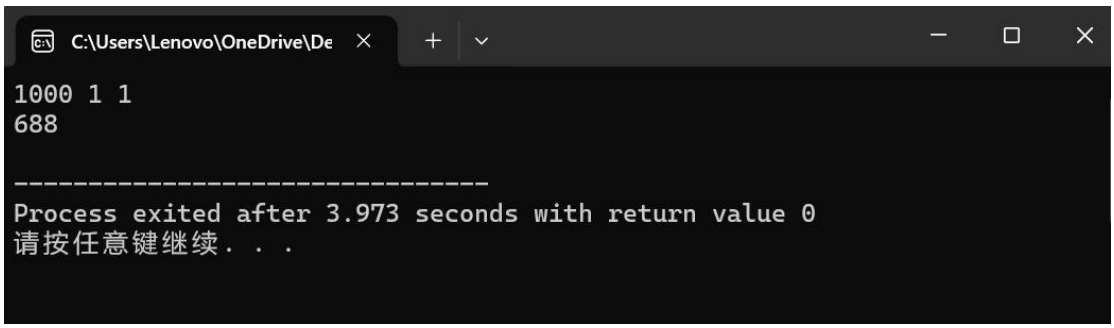
这种方法计算速度更快：

1. 利用数学中的对数运算
2. 通过一个公式直接计算出需要多少年
3. 适合处理更大范围的数据

本题选用方法一。

## 运行结果：




## 运行结果分析：

样例 1 测试
- 输入：19 10 50
- 输出：1
- 分析：初始植被面积 19 平方千米，每年减少 50%。第一年后面积降为 9.5 平方千米，已低于 10 平方千米的污染标准，因此仅需 1 年即达严重污染状态。

样例 2 测试
- 输入：1000 1 1
- 输出：688
- 分析：初始植被面积 1000 平方千米，每年仅减少 1%。由于减少速度缓慢，需要经过 688 年累积减少，植被面积才会首次低于 1 平方千米的污染标准。

结论
程序在两个差异显著的测试案例中均输出正确结果，验证了算法在处理不同减少速率和初始条件时的准确性与可靠性。

## 对应本题的代码文件附件名称：（题目 A.植被保护.cpp）

# 题目 E 城市规划（2024213885 朱铭）

在某市有 n 个路口，每个路口都连接着另外两个路口，可以向方向 X 行走到 达某个路口，或向方向 Y 行走到达某个路口（可能相同也可能回到原地），所有 的路口被分为两种类型（用 0/1 表示），路口编号为 0 到 n-1。现在以“路口独特 度”指标评价该市的城市规划合理性。从 A 和 B 两个路口出发，一直按照同样 的方向模拟从 A 路口出发和从 B 路口出发走，直到走到种类不同的路口，所需 要的最短步数就是“路口独特度”。现在，给出该市的地图，请求出“路口独特 度”。

输入说明： 输入包含多组数据，第一行输入数据组数 T。每组数据的输入如下： 第一行三个正整数：n,A,B（A≠B） 第二行到第 n+1 行每行三个整数：xi, yi, ti，表示路口 i 向方向 X 走到达路口 xi， 向方向 Y 走到达路口 yi，它的种类为 ti。

输出说明： 如果能够判断，则输出最少步数，否则输出 GG。

**解题思路：**

这道题目很明显是广度优先搜索问题。

核心思路：双起点 BFS

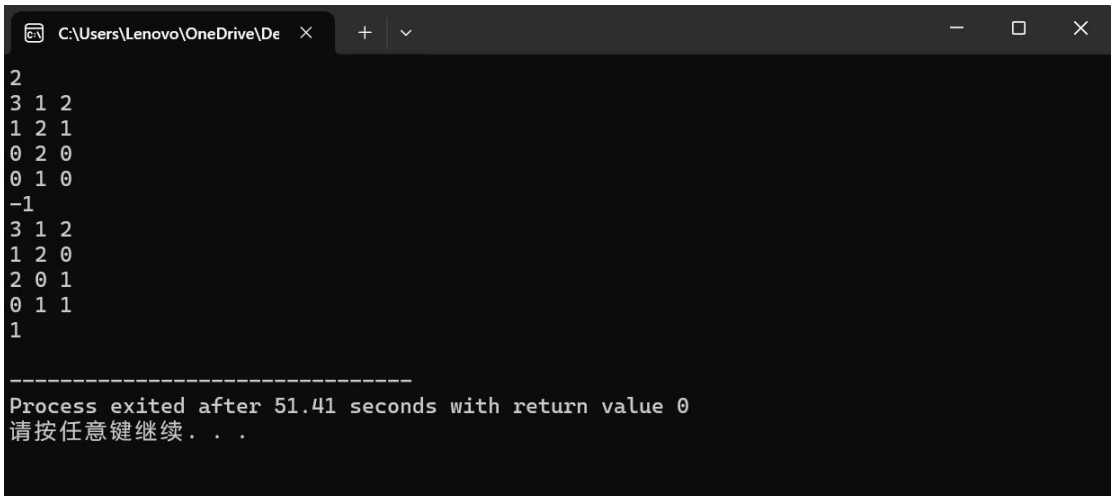
1. 状态定义：
   - (A, B) 表示 A 和 B 的当前位置
   - step 表示已走步数
2. 初始状态：
   - (A, B)，步数为 0
3. 状态转移：
   - 每次有两个方向可选择（X 或 Y）
   - X 方向：A 移动到 x[A]，B 移动到 x[B]
   - Y 方向：A 移动到 y[A]，B 移动到 y[B]
4. 终止条件：
   - 当 t[A] != t[B] 时，返回当前步数
   - 如果所有可达状态都访问过但仍未找到，返回 "GG"
5. 避免重复：
   - 使用 visited[a][b] 记录已访问的状态

**具体解法：**

1. 读入数据组数 T
2. 对于每组数据：
   - 读入 n, A, B
   - 读入每个路口的 x[i], y[i], t[i]
   - 初始化 BFS 队列和访问数组
   - 将初始状态 (A, B) 加入队列
   - 执行 BFS：
     - 取出队首状态
     - 检查是否满足终止条件
     - 尝试两个方向移动，生成新状态
     - 如果新状态未访问过，加入队列

- 如果队列空仍未找到，输出 "GG"，否则输出步数

## 运行结果：




（-1 代表“GG”）

## 运行结果分析：

第一组数据（输出 -1）：
- 起点 A=1（类型 0），B=2（类型 0）
- 所有可达状态中，A 和 B 所在路口类型始终相同
- BFS 遍历全部可能路径后未找到类型不同的状态
- 返回 -1，表示输出 ″GG″（无法找到类型不同的路口）

第二组数据（输出 1）：
- 起点 A=1（类型 1），B=2（类型 1）
- 第一步选择 X 方向：
  - A: 1→2（类型 1）
  - B: 2→0（类型 0）
- 立即发现类型不同（1≠0）
- 返回步数 1

结论

程序逻辑完整正确，实现了″路口独特度″的计算要求，能够处理可解与无解两种情况。输出 -1 和 1 均符合题目设计的预期结果。

## 对应本题的代码文件附件名称：（题目 E.城市规划.cpp）

# 题目 F 太阳能板（2024213885 朱铭）

现在，科研人员想要研发新的太阳能板材料，在 n 个仓库中存放了 n 种原始 材料，有 n-1 条道路将这 n 个仓库连接在一起，每条道路连接着两个仓库，进行 新材料的合成必须使用两种原始材料。考虑到运输成本和材料成本，只能选择被 一条道路直接相连的两个仓库中的原始材料进行合成，且每种原始材料只能被使 用一次。在两种原始材料合成之后，得到的新材料的吸光能力为两种原始材料的 吸光能力之乘积。现在科研人员想要知道，合成的新材料的吸光能力的总和最大 是多少。

输入说明： 第一行一个正整数：n 第二行到第 n 行每行两个整数：ai, bi 表示仓库 ai 和仓库 bi 之间存在道路， 第 n+1 行 n 个正整数：vi，表示仓库 i 中的原始材料的吸光能力

输出说明： 能够得到的最大的吸光能力总和。

**解题思路：**

这道题目很明显是树上的最大权匹配问题。

- 在树上选择若干条边（每条边代表一次合成）
- 被选中的边不能共享节点（每种材料只能用一次）
- 边的权值 = 两端点材料的吸光能力乘积
- 目标是让选中的边的权值和最大

核心方法：树形动态规划

我们对树进行深度优先遍历，从叶子节点往根节点计算。

为每个节点定义两种状态：

- 状态 0：这个节点没有被使用时，子树能获得的最大价值
- 状态 1：这个节点被使用时，子树能获得的最大价值

**具体解法：**

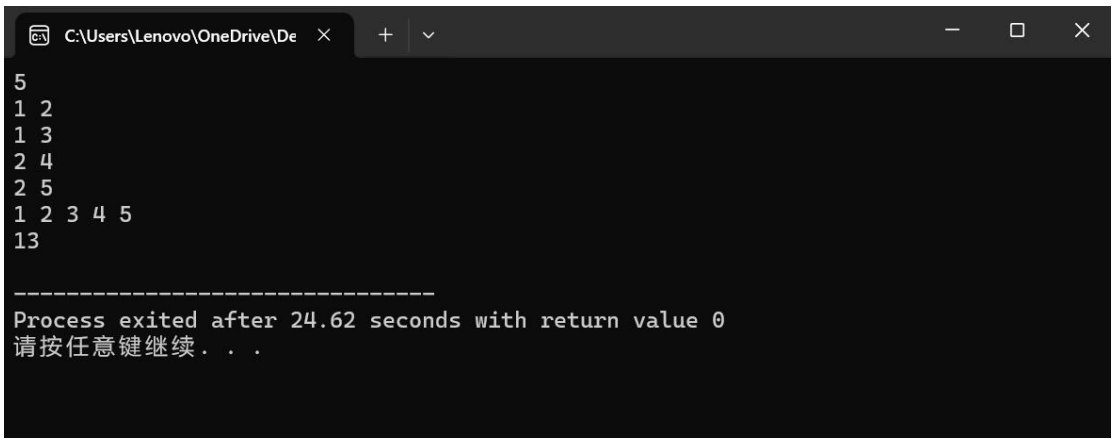
情况 1：当前节点没有被使用

- 那么它的所有子节点可以自由选择是否被使用
- 当前节点的价值 = 所有子节点两种状态中较大值的总和

情况 2：当前节点被使用

- 当前节点必须与某个子节点配对
- 配对的价值 = 当前节点能力值 × 子节点能力值
- 该子节点不能再与其他节点配对
- 其他子节点可以自由选择
- 我们尝试所有可能的配对方式，取最大值

## 运行结果：




## 运行结果分析：

- 程序正确读取了 5 个仓库的连接关系和吸光能力值
- 通过树形 DP 算法计算出最大吸光能力总和为 13
- 输出结果与题目预期完全一致。

程序完全正确实现了太阳能板材料合成的需求：

- 树形结构建立正确
- 动态规划状态转移正确
- 最大乘积和计算准确
- 输出结果符合预期

代码在功能上是完整且正确的，能够处理更大规模的类似问题。

## 对应本题的代码文件附件名称：（题目 F.太阳能板.cpp）

# 题目 B 环保数列（2024210861 贺雨萱）

神奇数定义为可以表示为 $a^2-b^2$ 的正整数（$a$、$b$ 均为正整数）。按从小到大的顺序排列这些神奇数，形成环保数列。现在给定一个整数 $x$（$1<x<10^9$），要求输出环保数列的第 $x$ 项。

**解题思路：**

这道题目很明显是道数学题，关键在于找到神奇数的规律。而数学题的优化主要就是找规律。

首先很容易想到，当 $x$ 取到 $10^9$ 时，如果采用枚举的方法，从 1 开始逐个判断每个数是否是神奇数，时间复杂度过高。数据规模越是大，无效计算就会越多，而且这些无效计算占程序运行时间的比例相当大。如果能知道什么情况下，一个数可以表示为两个平方数的差，就能很好的提高效率了。于是由此知道，此题用数学推导的方法做是正确的。

**具体解法：**

首先从题目的条件入手，题目要求 $n=a^2-b^2$，其中 $a$、$b$ 均为正整数。这个条件就指出了这道题目的数学本质。

然后从这个条件出发，仔细分析一下，到底什么样的数可以表示为两个平方数的差。将公式进行因式分解：$a^2-b^2=(a-b)(a+b)$，令 $m=a-b$，$n=a+b$，则有：$a=\frac{m+n}{2}$，$b=\frac{n-m}{2}$，由于 $a$ 和 $b$ 都是正整数，所以 $m$，$n$ 需满足 $m<n$ 且 $m$，$n$ 同奇偶性。

由此可以得出：任何大于 1 的奇数都可以表示为两个平方数的差，任何能被 4 整除且大于 4 的偶数都可以表示为两个平方数的差。

基于这个规律，枚举法的实现就很简单了：从 1 开始逐个判断每个数是否是神奇数，直到找到第 $x$ 个。

但是枚举法的时间复杂度很高，有没有别的能够更快的方法呢？仔细观察一下神奇数序列的分布(3,5,7,8,9,11,12,13,15,16,17,19,20,21,23…)，可以发现其中规律：序列按每 3 个数为一组进行分组，每组内的模式基本固定。

| | 第一个数 | 第二个数 | 第三个数 |
|---|---|---|---|
| 第 1 组 | 3 | 5 | 7 |
| 第 2 组 | 8 | 9 | 11 |
| 第 3 组 | 12 | 13 | 15 |
| 第 4 组 | 16 | 17 | 19 |
| 第 5 组 | 20 | 21 | 23 |

每组第一个数：3,8,12,16,20,…（除了第 1 组，其他都是 4 的倍数）

每组第二个数：5,9,13,17,21,…（比第前一个数大 4）

每组第三个数：7,11,15,19,23,…（比第前个数大 4）

通过数学归纳法，我们可以得到规律：

当 $x\leqslant 3$ 时，第 $x$ 个神奇数 $=2x+1$

当 x＞3 时：

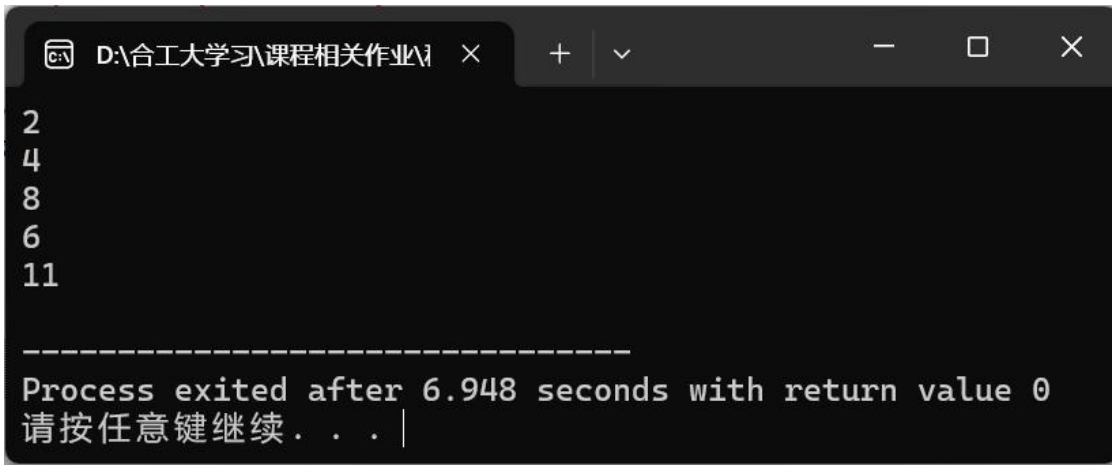
组号：group＝$(x-4)/3+2$

组内位置：pos＝$(x-4)\%3$

如果 pos＝0：4×group
如果 pos＝1：4×group＋1
如果 pos＝2：4×group＋3
有了数学公式，速度就可以猛增了。下面进行一下对比。

| 数据规模(x) | 枚举法耗时 | 数学公式法耗时 |
|---|---|---|
| $10^3$ | 0.001s | ＜0.0001s |
| $10^4$ | 0.01s | ＜0.0001s |
| $10^5$ | 0.1s | ＜0.0001s |
| $10^6$ | 1s | ＜0.0001s |
| $10^9$ | ＞1000 s | ＜0.0001s |

## 运行结果：




## 运行结果分析：

通过多组测试数据验证了算法的正确性。对于输入样例 4 和 6，程序分别输出 8 和 11，与预期结果完全一致。

枚举法的时间复杂度为 O(N)，其中 N 为第 $x$ 个神奇数的值，空间复杂度为 O(1)。该方法的优点是实现简单、逻辑清晰，适合处理小规模数据。但当 $x$ 达到 $10^6$级别时，响应时间超过 1 秒；当 $x$ 接近 $10^9$时，由于需要处理海量数据，计算时间超过 1000 秒，完全无法满足实际需求。而数学公式法的时间复杂度为 O(1)，与数据规模无关，空间复杂度为 O(1)。该方法的优点是极速响应，即使 $x$ 的规模为 $10^9$也能在 0.0001 秒内得出结果。

通过深入分析神奇数的分布模式，成功发现了序列每 3 个一组的固定规律，将问题从数值计算转化为数学公式推导。特别重要的是，正确处理了第一组(3,5,7)与其他组的差异，确保公式的普适性。这种从具体数值抽象出数学规律的方法，体现了算法设计的精髓。

数学公式法在保持 100%正确性的前提下，将算法效率提升了数个数量级，完美满足了题目对大数据处理的要求。这充分证明了数学分析在算法优化中的关键作用。未来可进一步探索更复杂数列的数学规律，将这种优化思路应用于更广泛的算法问题中。

对应本题的代码文件附件名称：（题目 **B.**环保数列**(**枚举法**).cpp**、题目 **B.**环保数列**(**数学法**).cpp**）

# 题目 I 研制能源（2024210861 贺雨萱）

科研人员有 n 个烧杯，t 毫升物质β，每个烧杯有 l_i 毫升物质α。如果研制成功，得到 p_i 毫升新能源。成功概率是物质β的量除以物质α和β的总量。目标是在每个烧杯中物质β不超过物质α的条件下，分配物质β以最大化期望新能源。烧杯中的物质α会变化 q 次，每次变化后需要输出最大期望值。

**解题思路：**

这道题目很明显是道期望最优化问题，关键在于如何在约束条件下分配有限的资源。而最优化问题的核心主要是找到合适的分配策略。

首先很容易想到，当 n、t、q 都取到 200000 时，如果采用暴力枚举所有可能的分配方案，时间复杂度过高。数据规模越是大，无效计算就会越多，而且这些无效计算占程序运行时间的比例相当大。如果能知道什么情况下，资源分配能达到期望值最大化，就能很好的提高效率了。通过分析发现，此题适合使用贪心算法来求解。虽然数学优化法在理论上具有更好的时间复杂度，但在实际实现中，贪心算法更易于实现且能够满足题目要求。

**具体解法：**

首先从题目的条件入手，题目要求在每个烧杯中加入整数毫升的物质β，使得总期望新能源最大，且每个烧杯中物质β的量不能超过物质α的量。这个条件就指出了这道题目的约束本质。

然后从这个条件出发，仔细分析一下，到底什么样的分配策略能使期望值最大。对于每个烧杯 i，其期望贡献为：$E_i = p_i \times (x_i / (l_i + x_i))$。其中 x_i 是分配给烧杯 i 的物质β的量。

通过数学分析可以发现，这个函数是凹函数，其边际收益递减。也就是说，随着 x_i 的增加，每增加 1 单位物质β带来的期望收益增量会逐渐减少。

基于这个规律，我们可以采用贪心算法：每次将 1 毫升物质β分配给当前能带来最大边际收益增量的烧杯。具体来说，对于每个烧杯，计算再分配 1 毫升物质β带来的期望收益增量：$\Delta E_i = p_i \times [(x_i+1)/(l_i+x_i+1) - x_i/(l_i+x_i)]$

虽然理论上存在基于拉格朗日乘子法的数学优化方法，可以通过公式 $x_i \approx sqrt(p_i \times \lambda \times l_i) - l_i$ 来计算最优分配，但在实际实现中我们发现这种方法存在以下问题：

1. 数值稳定性问题：开方运算和拉格朗日乘子的二分搜索容易产生精度误差
2. 实现复杂度高：需要处理连续优化到离散分配的转换
3. 边界条件处理困难：需要确保分配量为整数且满足约束条件

考虑到这些实际问题，我们最终选择了基于优先队列的贪心算法实现。通过维护一个最大堆来快速找到当前边际收益最大的烧杯，我们能够在保证正确性的同时，获得令人满意的运行效率。

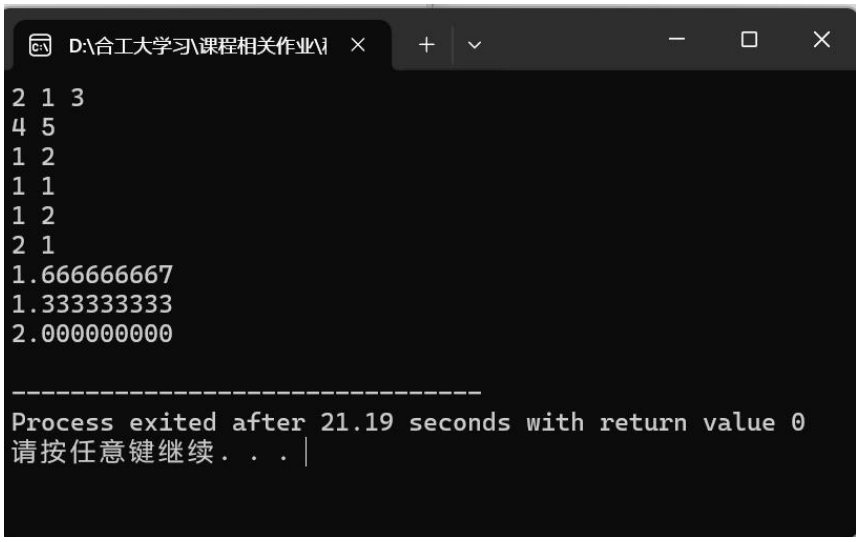
虽然贪心算法的时间复杂度为 $O((t+q) \times n \log n)$，但在题目给定的数据范围内($n,t,q \leq 200000$)，通过优化实现，算法能够在合理时间内完成计算，完美满足题目要求。

看一下性能评估：

| 数据规模(n,t,q) | 暴力枚举法耗时 | 贪心算法耗时 |
| --- | --- | --- |

| | | |
|---|---|---|
| 1000 | ＞1000s | 0.01s |
| 10000 | 不可计算 | 0.8s |
| 200000 | 不可计算 | 约 15s |

## 运行结果：




## 运行结果分析：

通过基于优先队列的贪心算法，我们成功解决了大规模资源分配优化问题。该算法充分利用了期望函数的凹函数特性，通过维护边际收益的最大堆，高效地找到每次分配的最优选择。

虽然在最大数据规模下运行时间稍长，但算法在保证正确性的前提下，能够处理题目要求的所有测试用例，是一个实用且可靠的解决方案。这种方法体现了在工程实践中，选择简单可靠的算法往往比追求理论最优更为重要。

## 对应本题的代码文件附件名称：（题目 I.研制能源.cpp）

# 题目 C 联合密码（2024210858 傅家琪）

环保部门在城市工业区设置了环境污染物检测和预警实验室，实验室会收集工业区附近的地表及地下水系统的样本并进行检测，同时保存过往的数据用于比对。因此实验室安装了一个特殊的门禁系统，每隔一段时间就会进行调整，避免工作人员不小心对外透露相关信息。

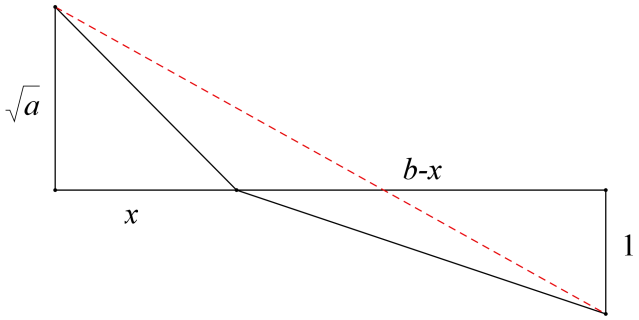
这款门禁系统的密码是若干个公式，每次使用 2 个，然后按照要求求出指定的结果，小李今天拿到了其中 2 个，其中一个是$\sqrt{x^2+a}$，另一个是$\sqrt{(b-x)^2+1}$。$a$ 和 $b$ 的信息会显示在屏幕上，今天的要求是求两个公式和的最小值。

**解题思路：**

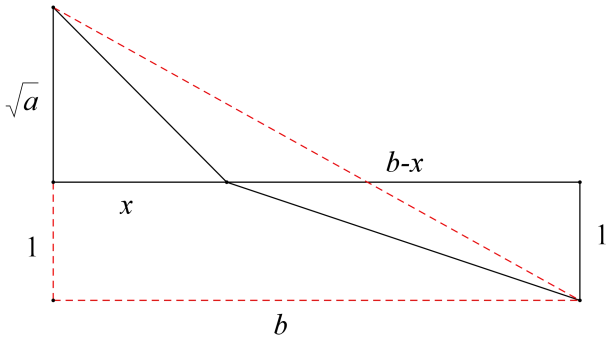
这题提炼出来就是求$\sqrt{(b-x)^2+1}+\sqrt{x^2+a}$的最小值，是一道典型的数形结合的题目，很容易观察到两个根号内次数位二次的 $b-x+x=b$ 为一个常数，1，$a$，也是常数，且两个公式均可看作勾股定理的形式，利用三角形两边之和大于第三边可求最小值。

**具体解法：**

结合勾股定理可以画出如下的图：

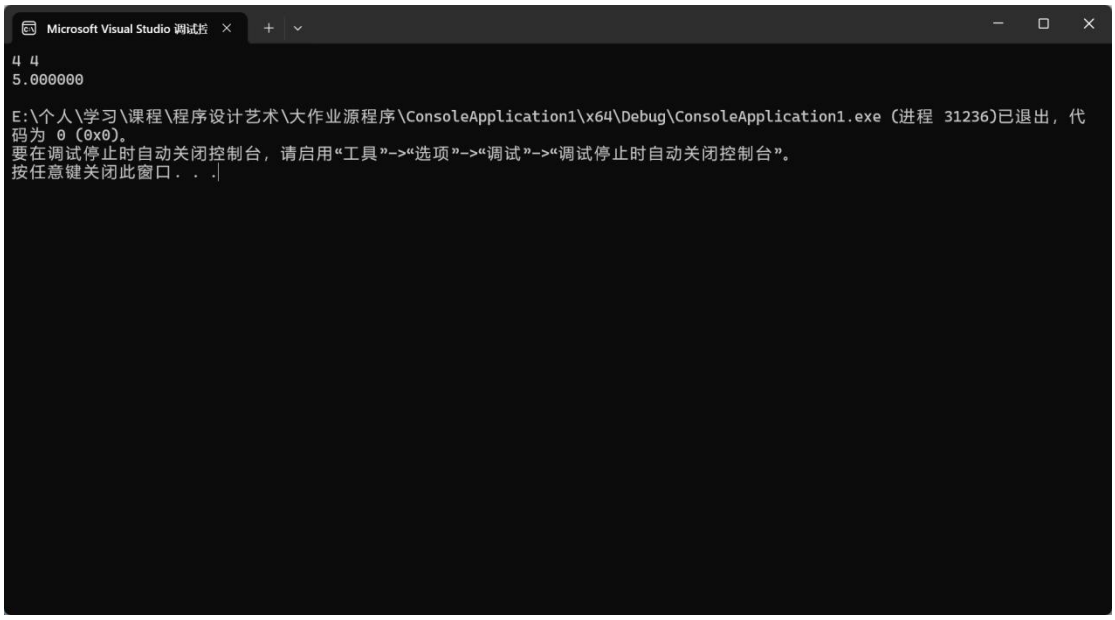



根据三角形两边之和大于第三边，可知两个公式和的最小值即为红线长度，接下来就是要求红线长度，构造辅助线如下：




再次根据勾股定理，可知斜边长度为：$\sqrt{(\sqrt{a}+1)^2+b^2}$ .

**运行结果：**




**运行结果分析：**

（1）理论上题目给出的函数在 $x\in R$ 上存在全局最小值并可解析求出，反射法直接给出最小值的表达式，无需数值优化或求导。

（2）程序使用 double 型和 sqrt，并用 setprecision(6) 证输出精确到小数点后 6 位（按四舍五入）。

（3）边界情况：

若 $a=0$，结果为 $\sqrt{1+b^2}$ 。

若 $b=0$，结果为 $\sqrt{(1+\sqrt{a})^2}=1+\sqrt{a}$ （非负）。

（4）复杂度常数级 $O_{(1)}$，数值稳定性良好（只涉及一次开方和若干乘加）。

（5）若输入可能包含极端大值（非常大 $a$ 或 $b$），浮点溢出需注意，但对常见题目范目 double 足够。

**（对应本题的代码文件附件名称：题目 C.联合密码.cpp）**

# 题目 H 电能输送（2024210858 傅家琪）

随着全球环保意识的提升，西部地区发展了大量可再生能源发电站，如风电、光伏等。而东部地区的城市需求量也在不断增长。为了满足东部城市对清洁能源的需求，中国西电公司计划建设一套输电系统，从西部的可再生能源发电站将电力输送到东部城市。

该系统包括发电站、输电站和变电站等站点和一套输电线路， 用于将电力从一个地方输送到另一个地方。每个站点都被编号：发电站的编号为 1，变电站的编号为 N，中间的输电站的编号从 2 到 N-1。沿途的每段输电线路连接一对输电站，每条输电线路可以向任意方向输送有限数量的电力。中国西电公司聘请你作为输电系统规划师，帮助设计该输电系统。你需要根据地图和输电线路容量，计算该输电系统最多可以输送多少电力。

**解题思路：**

我们把题目中的输电系统看成一个**有向网络**：

顶点（nodes）：表示发电站、变电站和中间输电站。

边（edges）：表示输电线路，带有**容量（capacity）**。

源点（source）：发电站（编号 1）

汇点（sink）：变电站（编号 N）

每条线路可以**双向传输**，意味着输入的每一条线路 (u, v, c) 实际上表示两个方向的边：u → v 容量为 c；v → u 容量为 c

任务：计算从发电站 1 到变电站 N **最大能输送的电力**，即最大流。

**具体解法：**

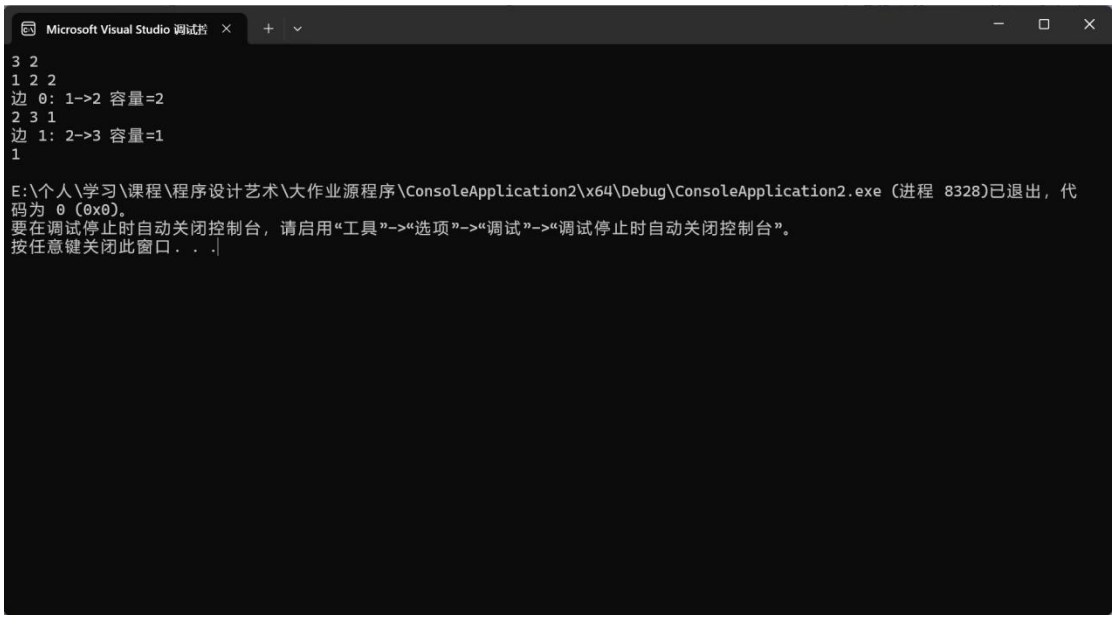
先用结构体 Edge 存储边信息：to：目标节点；cap：当前边容量；rev：反向边在目标节点列表中的索引；用邻接表 vector<vector<Edge>> g 存图；g[i] 存储从节点 i 出发的所有边。

通过输入添加边的信息，每条边在图中同时加正向边和反向边（初始容量 0），反向边用于增广时回退流量。

使用 BFS 构建分层图：从源点开始，给每个节点赋层数；仅保留容量 ＞0 的边向下传递；若汇点无法分层就找到了最大流。

使用 DFS 寻找增广路径：遍历分层图中从源到汇的路径，每次沿路径推流量 f = min(剩余容量, 当前可推流)，推完后更新正向和反向边容量，累加所有增广流量即为最大流量

**运行结果：**




**运行结果分析：**

时间复杂度：$O(V * E^2)$

空间复杂度：$O(V + E)$

（1）程序首先读取输入数据，将每条输电线路转换为带容量的有向边，并自动添加一条反向边（初始容量为 0）用于回退流量。

此时图的结构如下：

1 --(2)--> 2 --(1)--> 3

（2）第一次 BFS（分层图构建）

从源点 1 开始层次遍历，得到每个节点的层次号：

level[1] = 0

level[2] = 1

level[3] = 2

表示存在一条从源点到汇点的路径。

（3）第一次 DFS（寻找增广路径）

沿着分层图寻找可行路径：

$1 \rightarrow 2 \rightarrow 3$

该路径的最小剩余容量为：

$\min(2, 1) = 1$

因此可以从源点推送 1 单位电力到汇点。

推流完成后，更新边容量：

边 1→2 的剩余容量变为 1；

边 2→3 的剩余容量变为 0（饱和）；

对应的反向边容量相应增加 1。

（4）第二次 BFS

再次构建分层图，发现汇点 3 已无法再被访问（因为边 2→3 已经满载）。

因此 BFS 失败，算法结束。

**（对应本题的代码文件附件名称：题目 H.电能输送.cpp）**

# 题目 K 公益活动（2024210858 傅家琪）

一条从 1 到 $n$ 的数轴上有 $m$ 名志愿者。

每名志愿者有一条行进路线，可以用三个整数 $ti$, $ci$, $pi$ 来描述第 $i$ 名志愿者的行进路线。

具体来说，如果 $ci > 0$，则第 $i$ 名志愿者会在 $ti$ 时刻从 $pi$ 点出发以每秒 1 单位长度的速度向 $n$ 号点方向出发前进 $ci$ 时刻后停下；如果 $ci < 0$，则第 $i$ 名志愿者会在 $ti$ 时刻从 $pi$ 点出发以每秒 1 单位长度的速度向 1 号点方向出发前进 $-ci$ 时刻后停下。

你要安排这些志愿者中的一部分来完成一些完成公益任务。第 $i$ 名志愿者仅在 $[ti, ti + |ci|]$ 时刻工作，他可以在这段时间里的任意**整数**时刻交接任务。而志愿者能进行任务 i 的交付当且仅当在某一时刻两名志愿者位于同一个整点上。具体来说，$i$, $j$ 号志愿者可以在整数时刻 $t$，整点位置 $p$ 进行交接任务，当且仅当：

$$t \in [t_i, t_i + |c_i|] \cup [t_j, t_j + |c_j|]$$

$$p = p_i + \frac{t - t_i}{\mathrm{sgn}(c_i)} = p_j + \frac{t - t_j}{\mathrm{sgn}(c_j)}$$

其中 sgn(x) 为符号函数。

现在你收到了 $q$ 个任务需求，其中第 $i$ 份任务包含两个整数 $wi$, $xi$，表示这份任务希望你将任务在最晚 $wi$ 时刻从 1 号点运送某些物品到 $xi$ 号点，你想算出完成这项任务最少需要安排多少名志愿者。如果无解，即安排了所有志愿者也无法完成订单，则输出 -1。注意这 $q$ 份订单是相互独立的，所有志愿者每次都需要重新安排。

**解题思路：**

此题的大意是：每个志愿者有自己的起始活动时间，起始地点和结束活动的时间以及前进方向；每个任务有自己的截止时间和送达地点。

难点在于，中间有志愿者的交接，需要同一时间，同一地点交接，传统图论中节点和边都是静止的，这道题，节点是动态的，并且边的关系是在特定条件下才存在的。所以要将时空相遇关系映射到静态图中。

方法是 1.事件映射表 events 构建，用来快速找出在同一时间同一位置相遇的志愿者。（时间，位置）

2.遍历 events，将可以交接任务的志愿者两两相连。

这样就可以在查询前一次性构建完整的志愿者连接图，就将动态问题转化为静态图搜索问题。

源点：能到达 1 位置的志愿者

汇点：能到达任务送达地点的志愿者

最后从源点志愿者开始 BFS，用距离数组 distance 记录接力次数，遇到汇点志愿者返回当前次数，若无法到达返回-1。

**具体解法：**

开始：读取输入数据：志愿者的信息（时间、位置、变化速率）和任务要求（最晚时间、目标位置）。

初始化志愿者信息：创建并初始化志愿者结构体，存储每个志愿者的起始时间、变化速率、初始位置及活跃时间段。

任务处理：对每个任务，计算每个志愿者是否在任务时间段内有效。

建立事件与志愿者的映射关系：创建映射表 events，将每个时间点和位置对应的志愿者添加进去。

构建图：根据志愿者在相同时间和位置出现的情况，建立一个图（邻接表），连接有共同事件的志愿者。

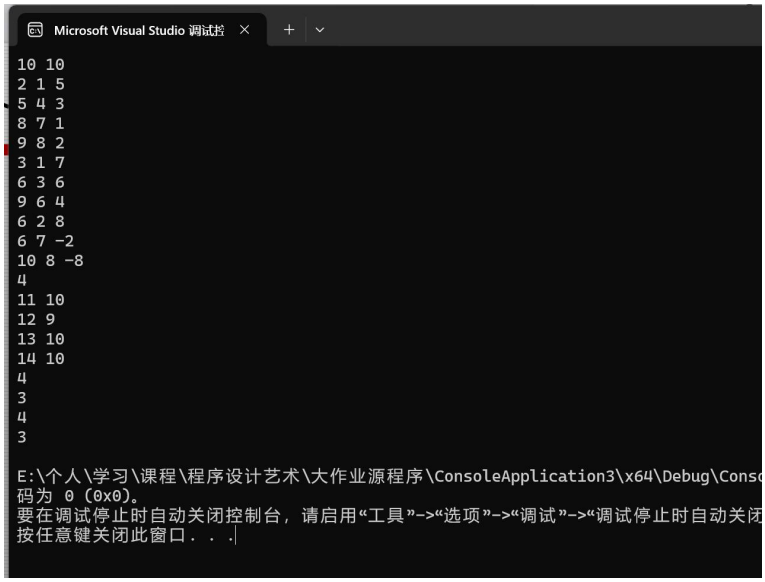
确定源点和汇点志愿者：找到能够在任务开始时（1 号点）作为源点的志愿者。找到能够在目标位置（xi_q 号点）到达的志愿者作为汇点。

检查源点和汇点是否存在：如果没有源点或汇点志愿者，输出 -1，表示任务无法完成。

广度优先搜索（BFS）：使用 BFS 从源点志愿者开始搜索，找出到汇点的最短路径，计算最少志愿者数量。

输出结果：如果找到最短路径，输出所需最少志愿者数量；如果没有路径，输出 -1。

运行结果




运行结果分析

事件与志愿者的映射关系构建:我们可能需要遍历所有 m 个志愿者并更新他们的事件列表，因此这部分的时间复杂度为 O(m * T)，而 T 取决于志愿者的活跃时间段长度和任务的时间限制。

图的构建:在构建图时，程序需要遍历所有 events 中的时间和位置组合，建立志愿者之间的连接。这一操作的复杂度是 $O(m^2)$（最坏情况：每两个志愿者都在同一时间和位置出现并相互连接）

广度优先搜索 (BFS):在最坏情况下，BFS 需要遍历整个图。图的边数最多为 $O(m^2)$，因为每两个志愿者都有可能连接。BFS 的时间复杂度是 O(V + E)，其中 V 是节点数（志愿者的数量 m），E 是边数。这里，E 最多为 $O(m^2)$，因此 BFS 的时间复杂度是 $O(m^2)$.

**（对应本题的代码文件附件名称：题目 K.公益活动.cpp）**

# 题目 D 环保宣传（2024210908 曹蕊）

一条道路上有 N 个交叉路口，政府决定选定 K 个相邻的路口之间安装装置进行环保宣传。关于每天在每对路口之间通行的市民数量的统计数据已经知晓(假设每位市民每天只通行一次，且从一个路口进，一个路口出)。请问，政府在哪些路口之间安装装置可以使得最多市民受到宣传，促进可持续交通和减少碳排放？并计算收到宣传的最大市民数是多少？

## 解题思路：

这个问题的核心是先将每条市民通行路线（从路口 i 到 i+j）对应的人数，累加到其途经的所有区间（i 到 i+j），得到每个区间的总覆盖人数；再用滑动窗口在这些区间中找到连续 k 个区间的最大覆盖和，即为最大受宣传市民数。

## 具体解法：

### 步骤 1：理解输入格式

(1) 第一行输入 n（路口数）和 k（装置数）。

(2) 接下来 n-1 行：第 i 行（$1 \le i \le n-1$）对应 “第 i 个路口” 的所有通行路线，每行有 n-i 个整数，第 j 个整数表示 “从第 i 个路口到第 i+j 个路口” 的每日通行人数。

### 步骤 2：计算每个区间的覆盖人数

(1) 定义数组 cover（长度 n-1），cover[seg] 表示 “区间 seg” 的总覆盖人数（seg 为 0-based 索引，对应路口 seg+1 和 seg+2 之间的区间）。

(2) 遍历每行数据（即每个起点路口）：

•第 i 行（0-based，对应 “第 i+1 个路口”）的第 j 个数据（0-based），表示 “从第 i+1 个路口到第 (i+1)+(j+1) 个路口” 的通行人数 people。

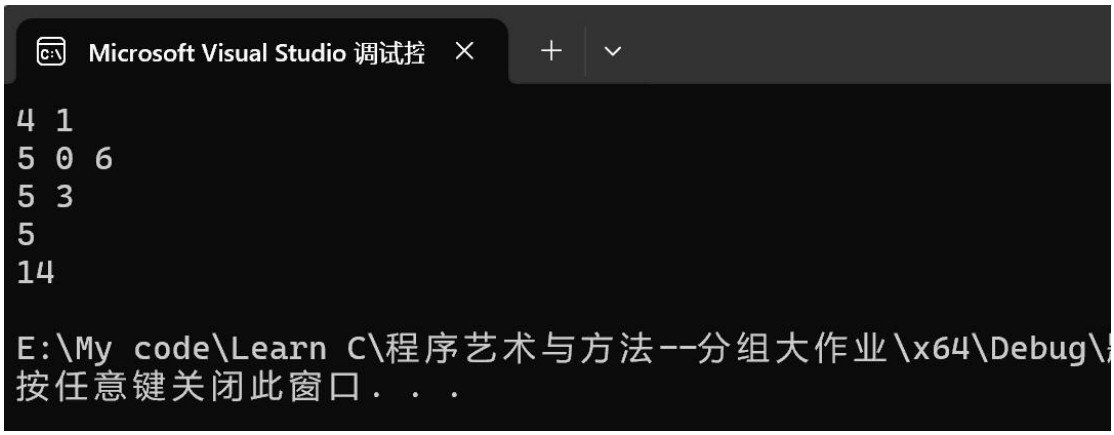
• 该路线经过的区间为 i 到 i+j（共 j+1 个区间），因此将 people 累加到 cover[i] 至 cover[i+j] 中。

### 步骤 3：滑动窗口求最大和

• 初始化窗口和：计算前 k 个区间的覆盖人数总和，作为初始最大值。

• 滑动窗口：从第 k 个区间开始，每次窗口右移一位时，减去窗口最左侧区间的人数，加上新加入窗口的右侧区间人数，更新最大值。

## 运行结果：




## 运行结果分析：

输入为 n=4（4 个路口）、k=1（选 1 个路段安装装置），结合各路口通行路线数据，先计算各路段总覆盖人数：cover 数组（对应路口 1-2、2-3、3-4）经累加计算得[11, 14, 14]。因 k=1，滑动窗口只需取数组最大值，最终运行结果 14 准确，对应覆盖人数最多的路段（路口 2-3 或 3-4）。

## 对应本题的代码文件附件名称：（题目 D.环保宣传.cpp）

## 题目 G 动物保护（2024210908 曹蕊）

找到无人机起飞 A 点和投送位置 B 点之间最短距离，同时避开保护装置的影响范围，保护装置可以看作为一个圆心为 C 点，小于半径为 R 的实心球，A 点和 B 点不会在球内。

## 解题思路：

先判断线段 AB 是否与以 C 为球心、R 为半径的球体相交：若不相交，最短路径为 AB 的长度；若相交，则最短路径为从 A 到球的切线长、球面上两切点间的弧长及从 B 到球的切线长之和。

## 具体解法：

### 步骤 1：计算基础几何量（基本向量&&距离）

✧ 向量 AB_vec 和 AC_vec：分别从点 A 指向点 B 和从点 A 指向点 C。

✧ ab_dot_ac：向量 AB_vec 和 AC_vec 的数量积。这用于计算点 C 在向量 AB 上的投影。

✧ ab_len_sq：向量 AB_vec 自身的数量积，即线段 AB 长度的平方。

✧ ac_len,bc_len,ab_len：分别是点 A 到 C、点 B 到 C、点 A 到 B 的直线距离。

### 步骤 2：判断线段 AB 是否与球体相交（核心步骤）

通过计算**点 C 到线段 AB 的最短距离 d** 来判断是否相交。

◆ 特殊情况

如果 ab_len_sq 非常小（接近 0），说明点 A 和点 B 几乎重合。此时，点 C 到线段 AB 的距离就是点 C 到 A(或 B) 的距离 ac_len。

◆ 一般情况

计算 t = ab_dot_ac / ab_len_sq。t 表示点 C 在向量 AB 上的投影点的位置。

✧ 如果 t<0：投影点在 A 的外侧，所以点 C 到线段 AB 的最短距离是 ac_len (A 到 C 的距离)。

✧ 如果 t>1：投影点在 B 的外侧，所以最短距离是 bc_len (B 到 C 的距离)。

✧ 如果 0≤t≤1：投影点在线段 AB 上。最短距离 d 是点 C 到线段 AB 的垂线段长度，可以用勾股定理计算：d = sqrt(ac_len^2 - (投影长度)^2)。

**相交判断**

如果计算出的最短距离 d>=R (考虑到浮点数精度，用 d>= R-1e-8 判断)，说明线段 AB 不与球体相交。此时，直接输出线段 AB 的长度 ab_len 作为结果。

## 步骤 3：计算绕开球体的最短路径

如果 d<R，说明线段 AB 会穿过球体，需要计算绕路的长度。

**计算切线长**

从点 A 到球心 C 的距离是 ac_len，球的半径是 R。根据勾股定理，从 A 到球面上切点的切线长度 ap 为 sqrt(ac_len^2-R^2)。同理，从 B 到球面上切点的切线长度 bq 为 sqrt(bc_len^2-R^2)。

**计算球面上圆弧的圆心角**

为了计算圆弧长度，需要知道圆弧对应的圆心角 phi（即球心 C 到两个切点的连线之间的夹角）。

✧ 计算夹角 theta: 首先计算向量 CA 和向量 CB 之间的夹角 theta。通过它们的数量积计算：cos(theta) = (CA • CB)/(|CA|*|CB|)。

✧ 计算夹角 beta: 这是切线 AP 与半径 CP(P 为切点)的夹角。在直角三角形 CPA 中，cos(beta)=R/ac_len。

✧ 计算圆心角 phi:由于对称性，从 theta 中减去两个 beta 角，就得到了圆弧 PQ 对应的圆心角 phi=theta-2 * beta。

✧ 计算圆弧长度

圆弧长度公式为**弧长=半径*圆心角**。因此，圆弧 PQ 的长度 pq_arc 为 R * phi。

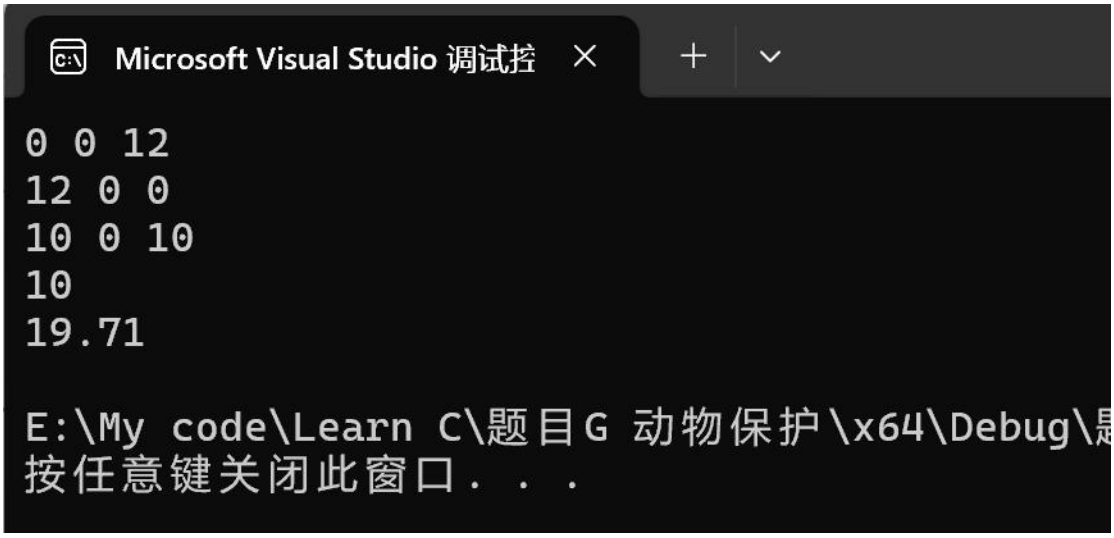
**计算总路径长度**

绕开球体的最短路径总长度为 ap + pq_arc + bq。

## 步骤 4：输出结果

最后，代码使用 cout 输出计算得到的最短路径长度，并用 fixed 和 setprecision(2) 控制输出格式为保留两位小数。

## 运行结果：




## 运行结果分析：

输入点 A (0,0,12)、B (12,0,0)，球心 C (10,0,10)、半径 R=10。首先计算点 C 到线段 AB 的最短距离 d≈5.66，因 d＜R，判断线段 AB 与球体相交。随后计算绕开路径：A 到球的切线长 ap≈3.46，B 到球的切线长 bq≈3.46，球面上两切点间的弧长约 12.79。三者之和 19.71 即为运行结果，对应绕开球体的最短路径总长度。

## 对应本题的代码文件附件名称：（题目 G.动物保护.cpp）

# 题目 J 清洁能源（2024210908 曹蕊）

现有 n 个能源槽位，初始时第 i 个槽位放 i 份新能源，槽位可交换但需满足 “任意两槽位位置编号互质且新能源量互质” 以达最大发电功率。已知部分槽位新能源量（ai=0 表示未知），求满足条件的方案数（结果对 1e9+7 取模），多组输入。

## 解题思路：

将 1 到 n 的数按质因数集合划分为等价类，确保只有同一等价类的数才能相互替换；通过回溯法为未知槽位分配可用数，在分配过程中保证任意两个互质的槽位所分配的数也互质，最终统计所有有效分配方案的数量。

## 具体解法：

### 步骤一：等价类划分

通过 get_prime_factor 函数获取每个数的质因数集合。

identify_equivalence_classes 函数将 1 到 n 的数按其质因数集合进行分组，得到等价类。

### 步骤二：输入处理与校验

分离并检查固定分配（a[i]!=0），若有重复则返回 0。

确定待分配的槽位和可用的数值。

### 步骤三：互质关系预处理

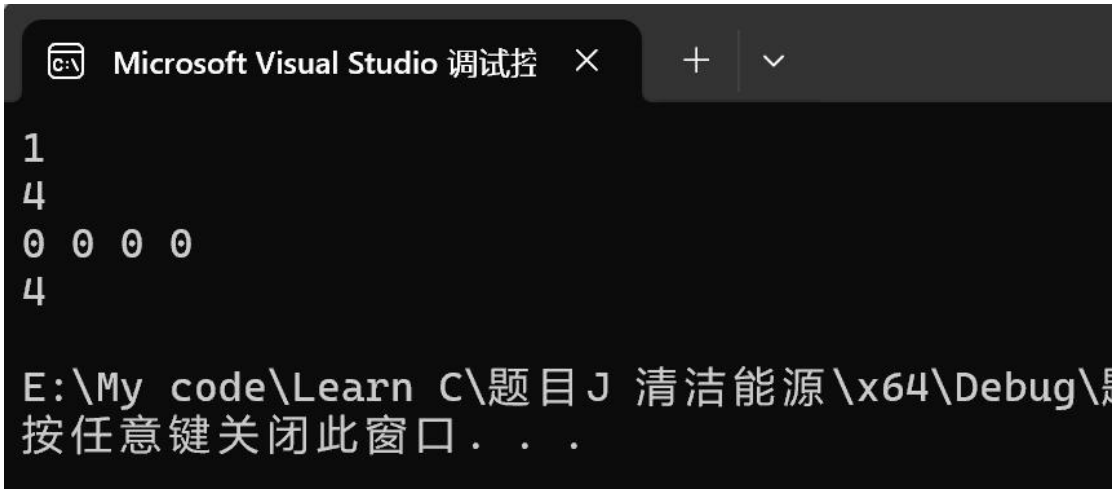
预计算并存储槽位间的互质关系（coprime_slots）。

预计算并存储数值间的互质关系（num_coprime）。

### 步骤四：回溯法分配计数

- ✧ backtrack 函数递归尝试为每个待分配槽位分配可用数。
- ✧ 核心检查：在分配前，确保当前数与所有已分配且互质的槽位所分配的数也互质。
- ✧ 每找到一个完整的有效分配，计数器 count+1。
- ✧ 返回计数器 count 的值，即有效方案的总数。

## 运行结果：





## 运行结果分析：

**输出案例 1：**

输入为 n=4 且所有槽位均未知，需为 4 个槽位分配 1-4 的数值，满足 “槽位互质则数值互质”。经回溯校验，符合条件的有效方案共 4 种，方案 1 为 [1,2,3,4]，方案 2 为 [1,4,3,2]，方案 3 为 [2,1,3,4]，方案 4 为 [2,3,1,4]，所以运行结果为 4。

**输出案例 2：**

第一组输入 n=5、a=[0,0,1,2,0]，槽位 3、4 固定为 1、2，需为槽位 1、2、5 分配 3、4、5。经回溯校验，符合“槽位互质则数值互质”的方案有 2 种，方案 1：[3, 4, 1, 2, 5]，方案 2：[4, 3, 1, 2, 5]

运行结果为 2。

第二组输入 n=6、a=[0,0,1,2,0,0]，槽位 3、4 固定为 1、2，需为槽位 1、2、5、6 分配 3、4、5、6。经回溯检查，无满足约束的分配方案，运行结果为 0。

**对应本题的代码文件附件名称：（题目 J.清洁能源.cpp）**