小马加编信息学教案(二十六)

二分答案

一. 课程内容

1. 题目引入
2. 算法分析

二. 知识讲解

1. 题目引入

> 经典问题：现在有n颗糖果，要把它们分成k个小朋友，拥有最多糖果的小孩至少有多少颗？

1. 1 思路一

> 我们可以从小到大枚举答案，即给每个小朋友一轮一轮地发糖果，每次给每个人发一颗，假设当前枚举到ans轮，即每个小朋友最多发ans颗糖果，最多可以分完$ans*k$颗，而现在我们有n颗糖果，即存在两种可能。
>
> 1. 若$n > ans*k$，则每个人发ans颗糖果都不能分完n颗，即ans为不合法答案，
> 2. 若$n <= ans*k$，则可以让每个人分得的糖果数都小于ans，即ans为合法答案。

> 所以对于枚举的判断是否合法时若`n <= ans * k`则ans为合法答案，否则为不合法答案。

> 枚举的过程可以分为两种情况考虑：
>
> 1. 若枚举到的ans不合法，那么证明答案要比ans大，继续枚举。
> 2. 若枚举到的ans合法，由于比ans小的答案均不合法，那么答案就是ans，结束算法。

> 这种做法，最多可能会枚举n个ans，所以时间复杂度是$O(n)$的。

1. 2 思路二

> 尝试一下优化思路一，减少不必要的枚举次数。

> 不难发现题目中的两个性质
>
> 1. 若某个ans为合法答案，那么1 ~ $ans-1$均为合法答案。
> 2. 若某个ans为不合法答案，那么$ans+1$ ~ n均为不合法答案。

> 那么回想上一节课**二分查找**中二分算法的应用。
> 问题是：在一个有序序列$a[]$中查找k的位置。
> 大概思路就是划出一个k可能存在的区间$[l,r]$，每次把这个区间平分为$[l,mid]$和$[mid+1,r]$两段，根据$a[mid]$和k的关系分成3种情况考虑：
>
> 1. $a[mid]<k$，则$[l,mid]$中所有数都小于k，只需考虑区间$[mid+1,r]$。
> 2. $a[mid]>k$，则$[mid+1,r]$中所有数都大于k，只需考虑区间$[l,mid]$。
> 3. $a[mid]=k$，找到k的位置，结束程序。

> 观察这道题目的性质，假如也将答案所处的范围设为$[l,r]$，一开始$l=1,r=n$，不断将答案所处区间划分为$[l,mid]$和$[mid+1,r]$，判断$ans=mid$时是否合法
>
> 1. 合法，那么$[l,mid]$中的数肯定都合法，答案至少为ans，继续考虑区间$[mid+1,r]$是否含合法答案。
> 2. 不合法，那$[mid+1,r]$种的数肯定不都合法，答案只可能在区间$[l,mid-1]$中。

> 核心代码
> `check(mid)`是一个判断$ans=mid$是否为合法答案的函数，若是则返回真，否则返回假。

```
	int l = 1, r = n, ans = n;
	while (l <= r) {
		int mid = (l + r) >> 1;
		if (check(mid)) {
			ans = mid;
			l = mid + 1;
			//第一种情况
		} else {
			r = mid - 1;
			//第二种情况
		}
	}
```

> 我们发现二分查找的思路同样也适用于这道题，只是我们并非二分位置，而是**二分答案**，那么同理，我们枚举ans的次数也从$O(n)$降低至$O(log_2^n)$，成功的优化了思路二的做法。

1. 3 思路三

> 其实这道题可以用$O(1)$的时间解决，并且比上面都简单。

> 根据思路一的方法，不难发现，我们只需要每个小朋友分到的糖果数尽量平均。
> 可以先让每个小朋友都分到一样的糖果数，并使这个数量最多，即`n / k`颗。剩下的糖果数就是`n % k`。假如剩余的糖果数大于0，那么有些小朋友就会多分到一颗。所以最后答案就是`n / k + ((n % k) > 0)`。

2. 算法分析

> 上面算法二中所用到的算法就叫做二分答案，虽然对于这道题没有算法三优，但是我们发现它能大大优化算法一的时间复杂度，对于某些含特殊条件的题目，会是十分好的选择，在算法竞赛中也是高频考点。

2. 1 适用条件

> 那么分析一下含有什么特殊条件时可以用到二分答案。

> 考虑二分答案的核心条件，对于当前答案所处区间$[l,r]$求出区间中点$mid=\frac{l+r}{2}$后划分区间。考虑题目需满足的性质
>
> 1. 若$ans=mid$是合法答案，则$[l,mid-1]$都要是合法答案才能保证不用再枚举左边的数，只处理区间$[mid+1,r]$
> 2. 若$ans=mid$不是合法答案，则$[mid+1,r]$都要是不合法答案才能保证不用枚举右边的数，只处理区间$[l,mid-1]$
>
> 总结一句话就是题目需要满足:
> **题目求的是合法答案的最小值或最大值，且若ans为合法答案，则$[1,ans-1]$均为合法答案，若ans为不合法答案，则$[ans+1,n]$均为不合法答案，即答案满足单调性，也可以称答案满足二分性。**

2. 2 使用思路

> 当题目满足二分性的时候，很多时候也不适合使用二分算法，比如这道题，可以通过算法三在$O(1)$的时间内得到答案。也有些题在二分答案$ans=mid$后无法快速判断ans是否为合法答案。

> 所以当答案满足二分性时，题目还需具备下面的性质。

> **题目答案很难由已知信息直接求出或无法在规定时间内求出时，需通过增加枚举的答案为已知条件，将原问题转化成判断某个答案是否合法的判定性问题。**
> 并且这个判定性问题是可以在规定时间解决的，那么我们基本确定这道题目可以通过二分答案的方法解决。

> 注意，当题目要求的是答案的最大值或最小值，并且答案满足二分性时，都要习惯的往二分答案方面想，因为只需花费$O(log_2^n)$的复杂度枚举，就可增加已知条件。然后通过上面的方法判断是否适合。

三. 经典例题

1. 分糖果

> 现在有n颗糖果，要把它们分成k个小朋友，拥有最多糖果的小孩至少有多少颗?
> 用二分实现

> 输入格式:
> 一行两个整数n,k，分别表示糖果数和小朋友数。
> $n,k\le 10^6$

> 输出格式:
> 一个整数表示答案。

样例输入	样例输出
11 5	3

2. 开根号

> 给定一个实数x，问x开根号是多少，精确到5位小数。
> 用二分实现。

> 输入格式
> 一行一个实数x。
> x在$double$范围内。

> 输出格式
> 一行一个使数表示x开根号的实数，精确到5位小数。

样例输入	样例输出
4	2.00000

3. 整数段分割

> 对于给定的一个长度为n的正整数数列a，现在要将其分成m段，并要求每段连续，且每段和的最大值X最小。问X是多少。

> 输入格式：
> 第一行两个整数n, m
> 第二行n个正整数表示数列a。
> $m \le n \le 10^5$
> $\sum a_i \le 10^9$

> 输出格式：
> 一个整数，表示X的最小值。

样例输入	样例输出
7 3 1 5 3 2 6 3 5	9

四. 提高巩固

1. 暴躁的牛

> 农夫有c头牛，n个隔间，c头牛很躁动，很容易相互打架，因此农夫想把它们分得越远越好，要你分配隔间使得相邻两头牛的距离越远越好，问你这c头牛分割的最小距离的最大值。

> 输入格式
> 第一行两个整数c, n表示有c头牛n个隔间。
> 接下来c行，每行一个整数d_i表示第i头牛的位置（按从小到大的顺序给出）。
> $n, c \le 10^5$
> $d_i \le 10^9$

样例输入	样例输出
5 3 1 3 4 6 7	1

2. 跳石头

> 一年一度的“跳石头”比赛又要开始了！
> 这项比赛将在一条笔直的河道中进行，河道中分布着一些巨大岩石。组委会已经选择好了两块岩石作为比赛起点和终点。在起点和终点之间，有n块岩石（不含起点和终点的岩石）。在比赛过程中，选手们将从起点出发，每一步跳向相邻的岩石，直至到达终点。
> 为了提高比赛难度，**组委会计划移走一些岩石，使得选手们在比赛过程中的最短跳跃距离尽可能长**。由于预算限制，

> 组委会至多从起点和终点之间移走M块岩石（不能移走起点和终点的岩石）。
> 问跳跃的最长长度是多少?

> 输入格式:
> 第一行三个整数l, m, n，分别表示起点到终点的距离，起点和终点之间的岩石数，以及组委会至多移走的岩石数。
> 接下来n行，每行一个整数，第i行的整数d_i，表示第i块岩石与起点的距离。这些岩石按与起点距离从小到大的顺序给出，且不会有两个岩石出现在同一个位置。
> $n, m \le 5 * 10^4$
> $l \le 10^9$

> 输出格式:
> 一个整数，即最短跳跃距离的最大值。

样例输入	样例输出
25 5 2 2 11 14 17 21	4

3. 砍树

> 伐木工人米尔科需要砍倒m米长的木材。这是一个对米尔科来说很容易的工作，因为他有一个漂亮的新伐木机，可以像野火一样砍倒森林。不过，米尔科只被允许砍倒单行树木。
> 米尔科的伐木机工作过程如下：米尔科设置一个高度参数h，伐木机升起一个巨大的锯片到高度h，并锯掉所有的树比h高的部分（当然，树木不高于h米的部分保持不变）。米尔科就行到树木被锯下的部分。
> 例如，如果一行树的高度分别为20，15，10和17，米尔科把锯片升到15米的高度，切割后树木剩下的高度将是15，15，10和15，而米尔科将从第1棵树得到5米，从第4棵树得到2米，共得到7米木材。
> 米尔科非常关注生态保护，所以他不会砍掉过多的木材。这正是他为什么尽可能高地设定伐木机锯片的原因。帮助米尔科找到伐木机锯片的最大的整数高度h，使得他能得到木材至少为m米。换句话说，如果再升高1米，则他将得不到m米木材。

> 输入格式
> 第一行两个整数n, m，分别表示树木的数量和需要木材的总长度。
> 第二行n个整数，第i个整数a_i表示第i棵树的高度。
> $n \le 10^5$
> $a_i, m \le 10^9$

> 输出格式:
> 一个整数h，表示砍树的高度。

样例输入	样例输出
5 20 4 42 40 26 46	36