算法与复杂性 第一次课程作业

UNIkeEN

2023 年 2 月 23 日

问题解答

1. 给出求解一元二次方程 $ax^2 + bx + c = 0$ 的算法。

> **解** 由定义，假设一元二次方程隐含 $a \neq 0$ 条件（即不退化为一元一次），且系数均为实数。
>
> 首先使用根的判别式判断方程根的情况，然后使用求根公式求解。**伪代码见算法1**
>
> 本题也可用牛顿迭代法，但对于一元二次方程而言，其最优方法即使用判别式法。

算法 1 公式法求解一元二次方程

输入: 实系数一元二次方程的系数 a, b, c

输出: 方程的根 x_1, x_2

$\Delta \leftarrow b^2 - 4ac$ ▷ 根的判别式
if $\Delta > 0$ **then** ▷ 方程有两个不同的实数根
  $x_1 \leftarrow \frac{-b+\sqrt{\Delta}}{2a}$
  $x_2 \leftarrow \frac{-b-\sqrt{\Delta}}{2a}$
else if $\Delta = 0$ **then** ▷ 方程有两个相等的实数根
  $x_{1,2} \leftarrow -\frac{b}{2a}$
else if $\Delta < 0$ **then** ▷ 方程有两个不同的复数根
  $x_1 \leftarrow \frac{-b+i\sqrt{-\Delta}}{2a}$
  $x_2 \leftarrow \frac{-b-i\sqrt{-\Delta}}{2a}$
end if
return x_1, x_2

2. 有一堆棋子，A 和 B 两人轮流从中拿 1-3 个，A 第一个拿，那么 A 如何确保自己不拿到最后一个棋子，给出相应的算法。

> **解** 这是一个巴什博弈问题。为使 A 确保不拿到最后一个棋子，A 最后一次拿完后必须只剩 1 个棋子，否则 B 可以拿取若干棋子后留下 1 个，则 A 不得不拿这最后一个棋子。
>
> 假设 B 第 n 次拿了 b_n 个棋子，A 下次拿 $a_{n+1} = 4 - b_n$ 个棋子，易证剩余棋子个数与 B 的决策无关。
>
> A 每次取棋子，需尽可能保持剩余棋子个数 N 满足 $N \equiv 1 \bmod 4$，**伪代码见算法2**。特别地，若 A 第一次取前，棋子总个数 N 已经满足 $N \equiv 1 \bmod 4$ 且 B 亦知此策略，A 将无法确保自己不拿到最后一个棋子。

算法 2 巴什博弈

输入: 本轮由 A 拿棋子时，剩余棋子个数 N

输出: A 本次拿取棋子的个数 t

$t \leftarrow (N-1) \bmod 4$

if $t \neq 0$ **then**

  return t ▷ A 此次拿 t 个棋子，保持剩余棋子个数 N 满足 $N \equiv 1 \bmod 4$

else

  $t \leftarrow$ 范围为 1-3 的随机数

  return t ▷ 本轮拿取 A 必定无法满足上述条件，则随意拿若干棋子

end if

3. 有一块长方形的巧克力，它由 $m \times n$ 个小块组成。你想要把它们全部掰开，但是每一步你只能拿起其中一块巧克力，沿着直线把它掰成两块。请证明，不管你用什么样的策略，把所有小块全部掰开所需要的步数都是相同的。

解 首先，由经验可得，将 $m \times n$ 的巧克力全部掰开，需要 $m \times n - 1$ 步，下证之：

方法一：双重数学归纳法

1. 显然，对于 $1 \times n$ 的巧克力，全部掰开需要 $n-1$ 步；对于 $m \times 1$ 的巧克力，全部掰开需要 $m-1$ 步，结论成立。
2. 假设对于 $(m+1) \times n$ 的巧克力和 $m \times (n+1)$ 的巧克力，结论成立，即全部掰开需要 $(m+1) \times n - 1$ 步和 $m \times (n+1) - 1$ 步。现在考虑 $(m+1) \times (n+1)$ 的巧克力，
   - 沿第 i 行（i 任意）掰一次，使之变为 $i \times (n+1)$ 与 $(m+1-i) \times (n+1)$ 的两小块巧克力，全部掰开分别需要 $i \times (n+1) - 1$ 和 $(m+1-i) \times (n+1) - 1$ 步。总步数为 $[i \times (n+1) - 1] + [(m+1-i) \times (n+1) - 1] + 1 = (m+1) \times (n+1) - 1$ 步。
   - 沿第 j 列（j 任意）掰一次，同理可证总步数为 $[(m+1) \times j - 1] + [(m+1) \times (n+1-j) - 1] + 1 = (m+1) \times (n+1) - 1$ 步。

综上所述，由双重归纳法得证，结论在 m、n 为任意正整数时成立。

方法二：第二数学归纳法 ＋ 双重归纳法

1. 显然，对于 $1 \times n$ 的巧克力，全部掰开需要 $n-1$ 步；对于 $m \times 1$ 的巧克力，全部掰开需要 $m-1$ 步，结论成立。
2. 假设对于 $m \times n$ 的巧克力，结论成立。现在考虑 $(m+1) \times n$ 的巧克力，

   假设第一次先掰成 $(m+1-k) \times n$ 和 $k \times n$，显然 $m+1-k$ 和 k 都在 $[1, m]$ 区间内，$[(m+1-k) \times n - 1] + (k \times n - 1) + 1 = (m+1) \times n - 1$；假设第一次先掰成 $(m+1) \times (n-k)$ 和 $(m+1) \times k$，继续掰，如果某次把 $m+1$ 这个维度掰开，回到归纳假设，假设一直掰直至变为 m 个 $1 \times n$ 的子块，也回到归纳假设

综上所述，由归纳法得证，结论在 m、n 为任意正整数时成立。

故对于一块由 $m \times n$ 个小块组成的长方形巧克力，无论用什么策略，把所有小块全部掰开所需要的步数都是相同的，均为 $m \times n - 1$。

4. 考虑数列 1，2，3，4，5，10，20，40，⋯，该数列从第 5 项起构成一个等比数列，证明任意一个正整数要么在数列中，要么可以表示成这个数列中的不同数之和。

解 易证，对于任意 $n \in \mathbb{N}^*$，存在唯一一组 (k, b)，其中 $k \in \mathbb{N}$，$b \in \{0, 1, 2, 3, 4\}$，使 n 满足

$$n = 5k + b \tag{1}$$

该数列的通项公式为

$$a_i = \begin{cases} i & n \leq 5 \\ 5 \cdot 2^{i-5} & n > 5 \end{cases} \tag{2}$$

对于 k，其可以转换为唯一的二进制表示，即 $k = \sum_{j=0}^{j} t_j \cdot 2^j$，其中 $t_j \in \{0, 1\}$

综上，对于任意 $n \in \mathbb{N}^*$，存在表示方式如下

$$\begin{aligned} n &= 5k + b & (3) \\ &= 5 \cdot \sum_{j=5}^{j} t_{j-5} \cdot 2^{j-5} + b & (4) \\ &= \sum_{j=5}^{j} t_{j-5} \cdot a_j + a_{n\ mod\ 5} & (5) \end{aligned}$$

其中 $k \in \mathbb{N}$，$b \in \{0, 1, 2, 3, 4\}$，$t_j \in \{0, 1\}$，当 n 被 5 整除时上式除去 a_0 项。

得证，任意一个正整数要么在数列中，要么可以表示成这个数列中的不同数之和（且加数不会重复）。

5. 有 10 个海盗抢得了 100 枚金币，每个海盗都能够很理智地判断自己的得失，他们决定这样分配金币：

   1. 按照强壮与否排序，其中最强壮的人为 10 号，以此类推，最瘦小的人为 1 号。
   2. 先由 10 号提出分配方案，然后由所有人表决，当且仅当等于或多于半数人（包括自己）同意时，方案才算被通过，否则他将被扔入大海喂鲨鱼；
   3. 如果 10 号被扔进大海，将由 9 号提方案，所有活着的人表决，当且仅当超过半数（包括自己）同意时，方案才算通过，否则 9 号同样将被扔入大海喂鲨鱼；
   4. 往下以此类推⋯⋯

海盗们都很精明，他们首先会尽量保住自己的命，其次在保住命的前提下都想分到尽可能多的金币，而且他们也很希望自己的同伴喂鲨鱼。假如你是那个 10 号海盗，你将怎样分配，才能既保住命，又能分到最多的金币？最多能分到多少呢？

如果还是 100 枚金币，但海盗的数量是 20，50，100，200，400 又该怎么样呢？

解 设 $a_i(j)$ 代表 i 号分配方案中分配给 j 号的金币数量

首先考虑只剩两个海盗（1、2 号）时，2 号可以将全部的 100 枚金币分给自己，然后给自己一票，此时满足“等于或多于半数人同意”。

引入 3 号，1 号知道如果 3 号被扔进大海，则会出现上述情况，1 号无法拿到金币。而 3 号为了活命，可以给 1 号金币以获取其投票，加上自己后共 2 票（超过半数），同时 3 号又想自己分到最多金币。则 3 号的分配方案将是 $\{a_3(1), a_3(2), a_3(3)\} = (1, 0, 99)$

引入 4 号，同理，4 号可以给 2 号金币以获取投票支持，同时自己分到最多金币。4 号的分配方案将是 $\{a_4(1), a_4(2), a_4(3), a_4(4)\} = (0, 1, 0, 99)$；

同理 5 号的分配方案将是 $\{a_5(1), a_5(2), a_5(3), a_5(4), a_5(5)\} = (1, 0, 1, 0, 98)$

猜想 1 当 $3 \le n \le 200$ 时，n 号海盗将提出如下方案，在保命的同时尽可能拿更多的金币。

$$c_n(i) = \begin{cases} 0 & i < n \text{ 且 } i \text{ 与 } n \text{ 奇偶性不同} \\ 1 & i < n \text{ 且 } i \text{ 与 } n \text{ 奇偶性相同} \\ 101 - \lceil \frac{n}{2} \rceil & i = n \end{cases} \tag{6}$$

证明 使用数学归纳法证明。

1. $n = 3$ 时的情况前已证明
2. 假设 $n = k$ 时猜想成立
   - 若 k 为奇数，则 k 号海盗可分得 $101 - \frac{k+1}{2}$ 个金币。且有 $\frac{k+1}{2} - 1$ 个奇数号海盗获得了 1 个金币，$\frac{k-1}{2}$ 个偶数号海盗没有获得金币。
   - 若 k 为偶数，则 k 号海盗可分得 $101 - \frac{k}{2}$ 个金币。且有 $\frac{k}{2} - 1$ 个偶数号海盗获得了 1 个金币，$\frac{k}{2}$ 个奇数号海盗没有获得金币。

   则当 $n = k + 1$ 时

   - 若 k 为奇数，$k + 1$ 为偶数。$k + 1$ 号海盗须得到另外 $\frac{k-1}{2}$ 个海盗的支持方可通过方案。因此，只需给 $n = k$ 时没有分得金币的 $\frac{k-1}{2}$ 个偶数号海盗分 1 个金币即可赢得他们的支持。即 $k + 1$ 为偶数时，

$$c_{k+1}(i) = \begin{cases} 0 & i < k+1 \text{ 且 } i \text{ 与 } k \text{ 奇偶性相同} \\ 1 & i < k+1 \text{ 且 } i \text{ 与 } k \text{ 奇偶性不同} \\ 100 - \frac{k-1}{2} = 101 - \lceil \frac{k+1}{2} \rceil & i = k+1 \end{cases} \tag{7}$$

   - 若 k 为偶数，$k + 1$ 为奇数。$k + 1$ 号海盗须得到另外 $\frac{k}{2}$ 个海盗的支持方可通过方案。因此，只需给 $n = k$ 时没有分得金币的 $\frac{k}{2}$ 个奇数号海盗分 1 个金币即可赢得他们的支持。即 $k + 1$ 为奇数时，

$$c_{k+1}(i) = \begin{cases} 0 & i < k+1 \text{ 且 } i \text{ 与 } k \text{ 奇偶性相同} \\ 1 & i < k+1 \text{ 且 } i \text{ 与 } k \text{ 奇偶性不同} \\ 100 - \frac{k}{2} = 101 - \lceil \frac{k+1}{2} \rceil & i = k+1 \end{cases} \tag{8}$$

综上归纳可得猜想 1 成立。

- $n = 201$ 时，201 号海盗必须获得除他自己以外的另外 100 名海盗的支持。因此，他可以分给 1-200 号海盗中 100 名奇数号海盗各 1 枚金币以保命，而他本人无法分得金币。

- $n=202$ 时，202 号海盗必须获得除他自己以外的另外 100 名海盗的支持。因此，他可以在 201 号决策时无法拿到金币的海盗（1-200 号海盗中偶数号海盗和 201 号海盗）中选择 100 名分配各 1 枚金币以保命，而他本人无法分得金币。

- $n=203$ 时，203 号海盗需要获得除自己外另外 101 名海盗的支持，但他只有 100 个金币，最坏只能获得 101 票（少于半数），**一定无法保住命**。

- $n=204$ 时，204 号海盗需要获得除自己外另外 101 名海盗的支持，虽然他只有 100 个金币，但 203 号为了保命也会支持 204 号（只要不轮到 203 号决策，他就是安全的），204 号最坏可以获得 102 票，他只需要在 202 号决策时无法分到金币的海盗中选择 100 位各分配 1 枚金币，就可以保命。

- $n=205$、$n=206$、$n=207$ 时，这几位号海盗需要获得除自己外另外 102 名海盗的支持，但他只有 100 个金币，最坏只能获得 101 票（少于半数），**一定无法保住命**。

- 当 $n=208$ 时，208 号海盗需要获得除自己外另外 103 名海盗的支持，虽然他只有 100 个金币，但 205、206、207 号为了保命也会支持 208 号（只要不轮到他们三个决策，他们三个就是安全的），208 号最坏可以获得 104 票，他只需要在 204 号决策时无法分到金币的海盗中选择 100 位各分配 1 枚金币，就可以保命。

猜想 2 当 $n>200$ 时，当且仅当 $n=200+2^k, k\in\mathbb{N}$ 时，存在可以通过的方案。

证明 使用数学归纳法证明

1. 由上述论述可知，当 $k=0,1,2,3$ 即 $n=201,202,204,208$ 时，存在可以通过的方案，且本人均无法得到金币。

2. 设 $k\in\mathbb{N}^*$。

   - 若 $n=200+2k$ 时，存在一个可以通过的方案，即 $100+k$ 个海盗支持了方案。则当 $n=201+2k$ 时，需要 $101+k$ 个海盗支持。除他自己和能用金币收买的 100 个海盗（共 101 人）以外，剩余 k 个人均会选择拒绝方案从而将他扔进大海。
   - 若 $n=200+2k$ 时，不存在一个可以通过的方案，即表示支持的海盗人数 $t<100+k$。则当 $n=201+2k$ 时，表示支持的海盗人数为 $t+1\le 100+k$，不足 $101+k$ 个。方案仍然不会被通过。

   所以当 $n>201$ 且 n 为奇数时，不存在能通过的方案。

3. 假设 $n=200+2^k$ 时，存在可以通过的方案。若 $n=n'>200+2^k$（由上述可知，n' 定为偶数）时，也存在可以通过的方案，且 $200+2^k<n<n'$ 时不存在可以通过的方案。则 $200+2^k$ 号至 $n'-1$ 号共 $n'-1-(200+2^k)$ 名海盗为了保命，都会支持 n' 号海盗。因此若有 n' 名海盗时存在可以通过的方案，定有

$$
\begin{aligned}
\left\lceil\frac{n'}{2}\right\rceil &= 100+1+(n'-1-(200+2^k)) \\
&= n'-2^k-100 \\
n' &= 2n'-2^{k+1}-200 \\
n' &= 200+2^{k+1}
\end{aligned}
$$

综上归纳可得猜想 2 成立。

答 代入数据可得

1. $n = 10$ 时，

$$a_{10}(i) = \begin{cases} 0 & i < 10 \text{ 且 } i \text{ 是奇数} \\ 1 & i < 10 \text{ 且 } i \text{ 是偶数} \\ 96 & i = 10 \end{cases} \tag{9}$$

2. $n = 20$ 时，

$$a_{20}(i) = \begin{cases} 0 & i < 20 \text{ 且 } i \text{ 是奇数} \\ 1 & i < 20 \text{ 且 } i \text{ 是偶数} \\ 91 & i = 20 \end{cases} \tag{10}$$

3. $n = 50$ 时，

$$a_{50}(i) = \begin{cases} 0 & i < 50 \text{ 且 } i \text{ 是奇数} \\ 1 & i < 50 \text{ 且 } i \text{ 是偶数} \\ 76 & i = 50 \end{cases} \tag{11}$$

4. $n = 100$ 时，

$$a_{100}(i) = \begin{cases} 0 & i < 100 \text{ 且 } i \text{ 是奇数} \\ 1 & i < 100 \text{ 且 } i \text{ 是偶数} \\ 51 & i = 100 \end{cases} \tag{12}$$

5. $n = 200$ 时，

$$a_{200}(i) = \begin{cases} 0 & i < 200 \text{ 且 } i \text{ 是奇数，和 } i = 200 \\ 1 & i < 200 \text{ 且 } i \text{ 是偶数} \end{cases} \tag{13}$$

6. $n = 400$ 时，不存在 $k \in \mathbb{N}^*$ 使 $200 + 2^k = 400$，因此 400 号海盗一定会被扔进大海。

6. 证明 $\log n = o(n^k)$，k 为正常数。

解 由定义可知，原题即证

$$\lim_{n \to +\infty} \frac{\log_m n}{n^k} = 0 \tag{14}$$

运用对数函数的换底公式与洛必达法则，由 k 是正常数，得

$$\begin{aligned}
\lim_{n\to+\infty}\frac{\log_m n}{n^k} &= \lim_{n\to+\infty}\frac{\ln n}{n^k\ln m} \quad (15)\\
&= \lim_{n\to+\infty}\frac{n^{-1}}{kn^{k-1}\ln m} \quad (16)\\
&= \lim_{n\to+\infty}\frac{1}{kn^k\ln m} \quad (17)\\
&= 0 \quad (18)
\end{aligned}$$

得证 $\log n = o(n^k)$，k 为正常数

7. 寻找单调递增函数 f(n) 和 g(n)，使得 f(n)=O(g(n)) 和 g(n)=O(f(n)) 都不成立。

解 构造函数 f(n) 和 g(n)，使 f(n)=O(g(n)) 和 g(n)=O(f(n)) 都不成立，即对于任意常数项 N，对所有 $n \geq N$，找不到常数 c_1, c_2 使 f(n)≤ c_1g(n) 和 g(n)≤ c_2g(n) 同时成立。

需使 f(n)、g(n) 交替式上升且令 h(n)=max{f(n)/g(n), g(n)/f(n)}，需使之单调递增，取 h(n)=n。

构造如下：

- $n = 1$ 时，f(n)=g(n)=1
- n>1 且 n 为奇数时，f(n)=ng(n)
- n>1 且 n 为偶数时，g(n)=nf(n)

表达式如下：

$$f(n)=\begin{cases} n! & ,n\text{为奇数} \\ (n-1)! & ,n\text{为偶数} \end{cases},\quad g(n)=\begin{cases} (n-1)! & ,n\text{为奇数} \\ n! & ,n\text{为偶数} \end{cases}$$

注：在本题讨论时间复杂度规模 n 时，默认其为正整数。

8. 假设解决同一个问题的两个算法 A1 和 A2 的时间复杂性分别为 $O(n^3)$ 和 $O(n)$，如果为这两个算法分别编写程序并在同样的环境下运行，算法 A2 的程序一定比算法 A1 的程序运行得快吗？为什么？

解 不一定。

影响程序运行时间的除了时间复杂度的主项，还包括较小项、系数和常数项。

可举反例如下，假设算法 A1 耗时 n^3，算法 A2 耗时 $3n$，当 $n = 2$ 时，有 $2^3 = 8 < 3\times3 = 9$，此时算法 A1 运行快于算法 A2。