實做能力加強

temmie

1. 位元運算
2. 字元轉換
3. 陣列的使用
4. 區間問題

位元運算

為什麼要學這個？

- 因為電腦的機制，位元運算會比一般運算更快一些
- 在**枚舉**這堂課上，我們會用到很多

進位制

- n 進位代表只用 n 以內的數字組成（如果超過 10 就用英文）
- 二進位的數字代表只用 0 和 1 組成

進位制

- n 進位代表只用 n 以內的數字組成（如果超過 10 就用英文）
- 二進位的數字代表只用 0 和 1 組成
- 如果一個式子有不同進位制的數字，則會用括弧備註在右下角
- 例如：$17_{(10)} = 10001_{(2)}$

二進位轉十進位

- 以 $10001_{(2)}$ 舉例
- 我們可以先把所有 2 的冪次列出來：2^4、2^3、2^2、2^1、2^0

二進位轉十進位

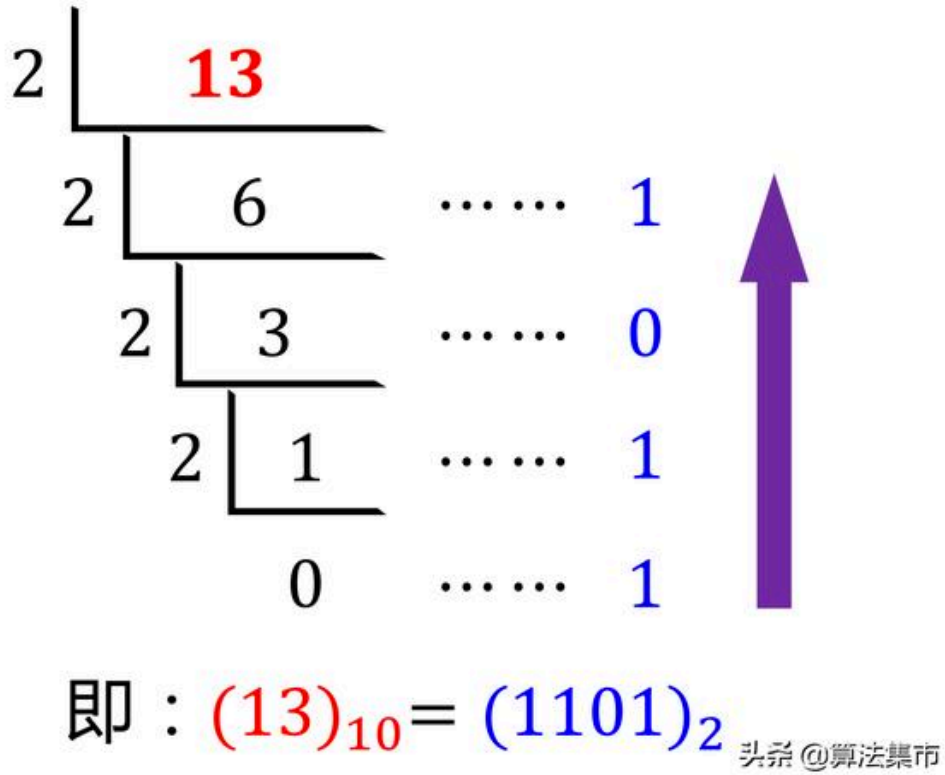
- 以 $10001_{(2)}$ 舉例
- 我們可以先把所有 2 的冪次列出來：2^4、2^3、2^2、2^1、2^0
- 和原本的二進位字串一一對應後相乘 1×2^4、0×2^3、0×2^2、0×2^1、1×2^0

二進位轉十進位

- 以 $10001_{(2)}$ 舉例
- 我們可以先把所有 2 的冪次列出來：2^4、2^3、2^2、2^1、2^0
- 和原本的二進位字串一一對應後相乘 1×2^4、0×2^3、0×2^2、0×2^1、1×2^0
- 將所有數字相加 $1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 17$

十進位轉二進位

- 詳細作法如下圖



例題

二進位制轉換

題目連結

給你一個十進位的正整數 n，輸出 n 的二進位

位元運算的類別

- 主要有 OR、AND、XOR、NOT
- 和一般的 or、and、not 不同的是，可以對數字做運算，而不是限制在布林

位元運算的類別

- 主要有 OR、AND、XOR、NOT
- 和一般的 or、and、not 不同的是，可以對數字做運算，而不是限制在布林

- *AND*：「&」表示，如果兩個 bit **都是** 1 就是 1，否則為 0
- *OR*：「|」表示，如果兩個 bit **至少**有一個 1 就是 1，否則為 0
- *XOR*：「^」表示，如果兩個 bit **恰有**一個為 1，否則為 0
- *NOT*：「~」表示，將 1 變成 0，0 變成 1

使用位元運算

- 難道用位元運算還要手動轉進位制？@@

使用位元運算

- 難道用位元運算還要手動轉進位制？@@
- 在 C++ 中，直接對兩個十進位的數字做就可以了
- $17_{(10)} \mid 5_{(10)} = 10001_{(2)} \mid 00101_{(2)} = 10101_{(2)} = 21_{(10)}$

例題

位元運算的大雜燴

題目連結

給你三個參數 x、y、z ，找到 $(x \oplus (y \mid z)) \oplus (x + y)$ 的最小值

其中 $\oplus$ 是 XOR 的意思，$\mid$ 是 OR 的意思

例題

位元運算的大雜燴

題目連結

給你三個參數 x、y、z ，找到 $(x \oplus (y \mid z)) \oplus (x + y)$ 的最小值

其中 $\oplus$ 是 XOR 的意思，$\mid$ 是 OR 的意思

- 參數數量是固定的，那就列出所有情況吧！
- 把六種組合各試一遍就可以了

更多位元運算

- 另外常用的位元運算有左移和右移

更多位元運算

- 另外常用的位元運算有左移和右移

- 左移：「＜＜」表示，代表將所有 bit 左移 n 位
- 右移：「＞＞」表示，代表將所有 bit 右移 n 位

更多位元運算

- 另外常用的位元運算有左移和右移

- 左移：「<<」表示，代表將所有 bit 左移 n 位
- 右移：「>>」表示，代表將所有 bit 右移 n 位

- 例如：$17_{(10)} << 1 = 10001_{(2)} << 1 = 100010_{(2)} = 34_{(10)}$
- 你知道嗎？實際上左移就是乘上 2^n（右移則相反）

字元轉換

ascii

- 我們都知道電腦是儲存 0 跟 1，並不能儲存字元
- 那要怎麼儲存字元呢？

ascii

- 我們都知道電腦是儲存 0 跟 1，並不能儲存字元
- 那要怎麼儲存字元呢？
- 只要把所有字母編碼就可以啦！
- 實際上我們用的字元都有一個編碼，稱為 ascii 編碼

ascii

- $[\ 0-9\ ] = 48 \sim 57$
- $[\ A-Z\ ] = 65 \sim 90$
- $[\ a-z\ ] = 97 \sim 122$
- 利用編碼的連續性，就有很多功能可以應用

例題

數字總和

題目連結

你有一個數字 n，請你求出所有位數的總和，保證 n 的位數 $\leq 10^5$

例題

數字總和

題目連結

你有一個數字 n，請你求出所有位數的總和，保證 n 的位數 $\leq 10^5$

- 字串很大，看起來沒辦法用 int 儲存，所以只能用 string 儲存
- 我們可以用 for 得到每個字元，並且**減去'0'**，就可以轉換成數字
- '0'(48)-'0'(48)=0，'1'(49)-'0'(48)=1
- '5'(53)-'0'(48)=5，'9'(57)-'0'(48)=9

例題

凱撒密碼

題目連結

給你一個字串 s，把每個字母向後移 3 位，例如 A $\rightarrow$ D，Z $\rightarrow$ C

例題

凱撒密碼

題目連結

給你一個字串 s，把每個字母向後移 3 位，例如 A $\rightarrow$ D，Z $\rightarrow$ C

- 我們可以將每個字母的值都加上 3
- 如果該字母超過'Z' 的話就減去 26

陣列的使用

區間問題