計算機程式設計二 Bonus 1 挑戰題講解

2185 - I2P(II)2020_Chen_bonus1

https://acm.cs.nthu.edu.tw/contest/2185/

截止日期：2020/12/06 18:40:00

參考答案釋出日期：2020/12/06 0:00:00

賴御誠　編著

Overview

- 13002 - the answer to life, the universe, and everything

13002 - the answer to life, the universe, and everything

難易度：★★★★★★★★★★

先備知識：前綴和(?)、費式數列

Description

- Knuckles 需要解一個謎題如下：
- 給定一個序列 F，F0 = 1、Fi = Fi-1 + Fi-2 與一個非負整數 n，他需要找到一個最小的 F1 ，使得有一個適合的 x 讓 Fx = n 成立
- 例如：n = 20，則最小的 F1 = 6
- 因為該 F 序列為 <1, 6, 7, 13, 20>，而 F4 = 20，並且我們無法找到一個更小的 F1 使得 n 出現在 F 序列之中

Input/Output

- 輸入包含至多 100 筆測資，對於每筆測資僅有一個整數 n
- 1 <= n <= 10^18

- 對於每筆測資，輸出一個整數，代表最小的非負整數 F1，且 n 會出現在 F 序列中

Input/Output

- 0
- 1
- 20
- 2147483647
- 765237384630454014
- 554
- 1221305
- 48367341873
- 731652777315453
- 997974924660318991

- 0
- 0
- 6
- 1073741823
- 765237384630454013
- 42
- 42
- 42
- 42
- 42

Hint by TAs

- 不要看到費式數列就想用矩陣快速冪，這題不需要這個也做得出來
- https://acm.cs.nthu.edu.tw/problem/10322/

透過展開與矩陣的性質，我們可以再次改寫費式數列：

$$\begin{bmatrix} fib_{n+1} \\ fib_n \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n \times \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Brain Storming

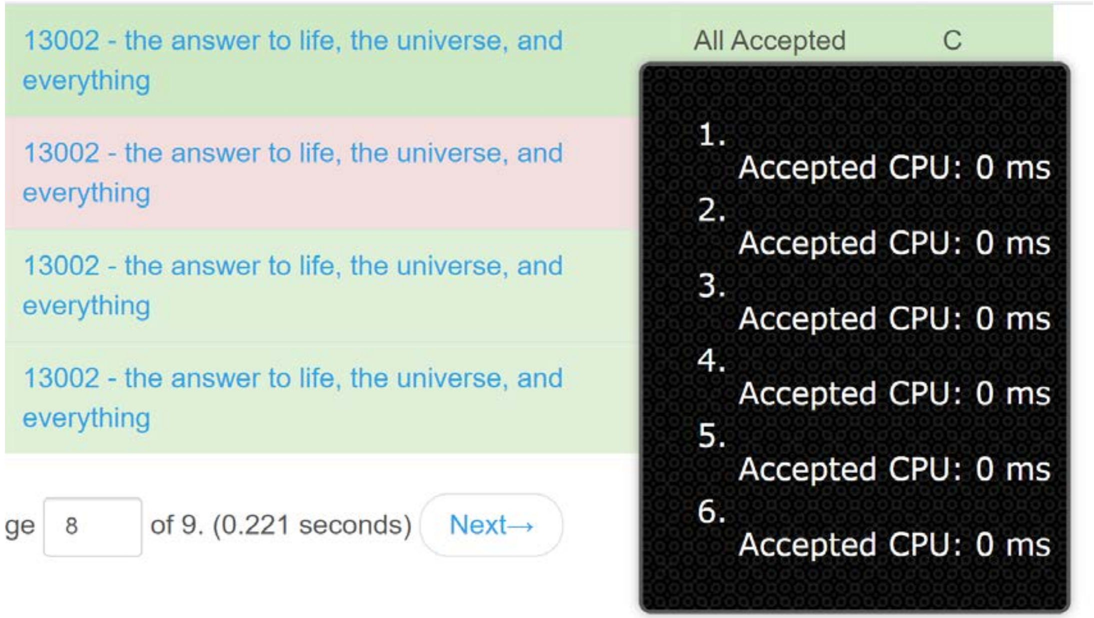
1	0	1	1	2	3	5	8	13	21	34	55	89	144	233	377
1	1	2	3	5	8	13	21	34	55	89	144	233	377	610	987
1	2	3	5	8	13	21	34	55	89	144	233	377	610	987	1597
1	3	4	7	11	18	29	47	76	123	199	322	521	843	1364	2207
1	4	5	9	14	23	37	60	97	157	254	411	665	1076	1741	2817
1	5	6	11	17	28	45	73	118	191	309	500	809	1309	2118	3427
1	6	7	13	20	33	53	86	139	225	364	589	953	1542	2495	4037
1	7	8	15	23	38	61	99	160	259	419	678	1097	1775	2872	4647
1	8	9	17	26	43	69	112	181	293	474	767	1241	2008	3249	5257
1	9	10	19	29	48	77	125	202	327	529	856	1385	2241	3626	5867

Brain Storming

- 由題目可見這個 F 序列很顯然是費式數列，且為遞迴關係式
- 因測資有高達 100 多筆，也就是需要先建立好 n 所有可能性的最小 F1 值
- 但是我們不可能在 C 語言建立一個高達 10^18 的陣列，這會讓記憶體高達 4GB 以上，已經超過 32 位元作業系統所能支撐的限度
- 使用前綴和？但是費式數列本身就是前綴和XD，且建表曠日廢時
- 沒有頭緒，那就先觀察其他人的耗時來決定需要用甚麼方法

Brain Storming

- 觀察其他 AC 的高手的耗時來判斷應用何種方法？
- 發現所有測資皆為 0ms
- 代表無論測資多大都能在一瞬間完成，也就是我們說的 O(1) 複雜度
- 這告訴我們可能存在公式解



Brain Storming

- 若存在公式解，那就必須要觀察序列或者上網找人家整理的結果
- 去維基百科、StackOverflow、GeekforGeeks 還是 Google 找看看關於斐波那契數列的特性吧
- https://en.wikipedia.org/wiki/Fibonacci_number
- https://math.stackexchange.com/questions/1871915/how-to-find-the-root-value-of-a-fibonacci-sequence-from-two-consecutive-values
- https://www.geeksforgeeks.org/print-first-n-fibonacci-numbers-using-direct-formula/

Brain Storming

- 經過一番尋找，看到了狹義費式數列的公式解 – Binet's formula
- 但….這個只能解 F0 = 1 與 F1 = 1 的 Fn 項
- 那麼，會不會跟檢查費式數列有關呢？
- https://www.geeksforgeeks.org/check-number-fibonacci-number/
- 若 (5*n^2+4) 或者 (5*n^2-4) 為一個完全平方數，則該數為費式數字
- 但，這個好像沒用XD，帶進去 Testcase 看看，沒用…
- 也許該找看看變動初始值的費式數列公式解？

Brain Storming

- https://math.stackexchange.com/questions/3314911/adjust-the-fibonacci-function-for-different-initial-conditions
- 若 G1 = a 且 G2 = b，則 Gn = aFn + bFn-1
- 則求 n 的公式解如下：

$$n = \left\lfloor \log_\varphi\left(\frac{G_n}{50} \cdot \sqrt{5} + \frac{1}{2}\right)\right\rfloor - 1.$$

Brain Storming

- 這個題目是對的，變動初始值，但這個公式需要已知 G1 與 G2 求 n 為多少，一方面我們不在乎 n 是多少，初始值也是未知
- 更不用說在程式碼裡面要把數學式建模是一個極具挑戰性的事情
- 感覺我們好像繞了一大圈，但是卻無所獲
- 網路上似乎沒有針對這個題目做討論QQ
- 那麼，那些人是怎麼做出來的呢？
- 只好回去觀察費式數列找規律…

Brain Storming

1	0	1	1	2	3	5	8	13	21	34	55	89	144	233	377
1	1	2	3	5	8	13	21	34	55	89	144	233	377	610	987
1	2	3	5	8	13	21	34	55	89	144	233	377	610	987	1597
1	3	4	7	11	18	29	47	76	123	199	322	521	843	1364	2207
1	4	5	9	14	23	37	60	97	157	254	411	665	1076	1741	2817
1	5	6	11	17	28	45	73	118	191	309	500	809	1309	2118	3427
1	6	7	13	20	33	53	86	139	225	364	589	953	1542	2495	4037
1	7	8	15	23	38	61	99	160	259	419	678	1097	1775	2872	4647
1	8	9	17	26	43	69	112	181	293	474	767	1241	2008	3249	5257
1	9	10	19	29	48	77	125	202	327	529	856	1385	2241	3626	5867

Brain Storming

- 看起來有些行有奇數偶數的關係，且似乎有上方加左方等於自己的趨勢
- 但是這也僅限說 F1 本身為費式數列的某一項時才會成立，有些 F1 產生的根本就完全脫節
- 似乎還是找不到可循的規律…

直到…你突然發現 拆 解

事情才有了曙光

Brain Storming

1	0	1	1	2	3	5	8	13	21	34	55	89	144	233	377
1	1	2	3	5	8	13	21	34	55	89	144	233	377	610	987
1	2	3	5	8	13	21	34	55	89	144	233	377	610	987	1597
1	3	4	7	11	18	29	47	76	123	199	322	521	843	1364	2207
1	4	5	9	14	23	37	60	97	157	254	411	665	1076	1741	2817
1	5	6	11	17	28	45	73	118	191	309	500	809	1309	2118	3427
1	6	7	13	20	33	53	86	139	225	364	589	953	1542	2495	4037
1	7	8	15	23	38	61	99	160	259	419	678	1097	1775	2872	4647
1	8	9	17	26	43	69	112	181	293	474	767	1241	2008	3249	5257
1	9	10	19	29	48	77	125	202	327	529	856	1385	2241	3626	5867

Brain Storming

- 我們把某幾行的數字不做總和，用拆解的方式看看
- 分析一下組成元素與 F0 F1 之間的關係

1	1	2	3	5
2	3	5	8	13
3	5	8	13	21
4	7	11	18	29
5	9	14	23	37
6	11	17	28	45
7	13	20	33	53
8	15	23	38	61
9	17	26	43	69
10	19	29	48	77

Brain Storming

- 發現了奇怪的規律！
- 那我們再看看如果今天是 F0 = 1 且 F1 = x 呢？

1*1+0*1	1*1+0*2	1*2+0*3	1*3+0*5	1*5+0*8
1*1+1*1	1*1+1*2	1*2+1*3	1*3+1*5	1*5+1*8
1*1+2*1	1*1+2*2	1*2+2*3	1*3+2*5	1*5+2*8
1*1+3*1	1*1+3*2	1*2+3*3	1*3+3*5	1*5+3*8
1*1+4*1	1*1+4*2	1*2+4*3	1*3+4*5	1*5+4*8
1*1+5*1	1*1+5*2	1*2+5*3	1*3+5*5	1*5+5*8
1*1+6*1	1*1+6*2	1*2+6*3	1*3+6*5	1*5+6*8
1*1+7*1	1*1+7*2	1*2+7*3	1*3+7*5	1*5+7*8
1*1+8*1	1*1+8*2	1*2+8*3	1*3+8*5	1*5+8*8
1*1+9*1	1*1+9*2	1*2+9*3	1*3+9*5	1*5+9*8

Brain Storming

	1+0*x	0+1*x	1+1*x	1+2*x	2+3*x	3+5*x	5+8*x	8+13*x
F0 = 1	1	0	1	1	2	3	5	8
F1 = x	0	1	1	2	3	5	8	13

- 可以發現 F0 與 F1 的參數序列似乎存在著一定規律
- 而且這個規律不就是狹義的費式數列嗎？！
- 也就是說如果能夠取得狹義的費式數列參數列表，我們就可以從 Fn 直接得知 x 的值！

Brain Storming

- 可以得到我們的算式 x = (n-Fn-1)/Fn
- 當 Fn 與 Fn-1 跟 n 已知時，可以直接知道 x 為多少
- 但是我們仍然不知道作為參數的費式數列需要建多長？
- 其實只要能夠覆蓋 10^18 就可以嚕

Brain Storming

- 以下是費式數列的各項列表：
- 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, 514229, 832040, 1346269, 2178309, 3524578, 5702887, 9227465, 14930352, 24157817, 39088169, 63245986, 102334155, 165580141, 267914296, 433494437, 701408733, 1134903170, 1836311903, 2971215073, 4807526976, 7778742049, 12586269025, 20365011074, 32951280099, 53316291173, 86267571272, 139583862445, 225851433717, 365435296162, 591286729879, 956722026041, 1548008755920, 2504730781961, 4052739537881, 6557470319842, 10610209857723, 17167680177565, 27777890035288, 44945570212853, 72723460248141, 117669030460994, 190392490709135, 308061521170129, 498454011879264, 806515533049393, 1304969544928657, 2111485077978050, 3416454622906707, 5527939700884757, 8944394323791464, 14472334024676221, 23416728348467685, 37889062373143906, 61305790721611591, 99194853094755497, 160500643816367088, 259695496911122585, 420196140727489673, 679891637638612258, 1100087778366101931

Brain Storming

- 可以觀察到當費式數列在大約 90幾項就已經超越 10^18
- 而我們的算式 x = (n-Fn-1)/Fn 需要先保證 n >= Fn-1 才能夠讓 x 為非負整數就好
- 且因為當 Fn-1 與 Fn 皆越小的時候，會使 x 越小，也就是題目要求的 F1 越小越好
- 所以我們需要從最後一個最大項開始找回來，如果都找不到適合的值就會落入最糟的狀況
- 最糟的狀況其實就是 F1 = n-1 時的 F2 項 = 1*1+1*x

Brain Storming

- 還有一個沒有討論的數列就是 F1 = 0 時，因為這個數列會先變大再變小
- 當遇到 n = 0 時掉到最糟狀況時會變成 n-1 也就是 -1，但是實際上答案是 0
- 直接當作特殊狀況處理

Design Flow

- 先建立參數用的費式數列矩陣 F ，起始值為 F0 = 1 與 F1 = 2，並且建立到 10^18 為止
- 開始讀入 n 直到 EOF，若 n = 0 則直接輸出 0，反之為其他數字時，則從第 85 項(或者任意大於 10^18 的數字都可以)開始往回推，直到 n 大於 0 為止
- 先判斷 n 是否大於等於 Fn-1，需要先是非負整數做除法才有意義
- 若 n-Fn-1 可以被 Fn 整除，則算出該 F1 為多少並輸出，就可以跳出迴圈
- 若 Fn 與 Fn-1 跑到最後一組組合都沒有辦法整除(也就是 index = 0)，則直接以最糟狀況處理，也就是輸出 n-1