Нека с **T[n]** означим родителите на един търтей от n-то коляно, а с **B[n]** родителите на една пчела от n-то коляно.

От условието на задачата получаваме следните рекурентни връзки:

**1) T[n] = B[n-1]**

**2) B[n] = T[n-1] + B[n-1]**

1) се получава, защото родителите от n-то коляно на един търтей са родителите от (n-1)-во коляно на неговата майка (пчела). Като прост пример можем да разгледаме родителите на един търтей от 2-ро коляно, т.е. неговите баби и дядовци, т.е. майката и бащата на майка му, т.е. родителите от 1-во коляно на майка му.

2) се получава с аналогични разсъждения, с разликата, че пчелата има и майка и баща.

От горните две уравнения можем да изведем следното уравнение за родителите на търтея:

**Т[n] = B[n-1] =** (от равенство 2) )

**= T[n-2] + B[n-2] =** (от равенсво 1) приложено към B[n-2] )

**= T[n-2] + T[n-1]**

По условие **T[0] = 1**, **T[1] = 1**

Другояче казано, това са числата на Фибоначи.

При дадените ограничения не е трудно да се направи линейно решение с динамично програмиране, за намирането на първите 105 числа на Фибоначи, след което при заявка да се извежда търсеният елемент от масива.

Трябва да се направи едно съображение – модулът, по който се търсят числата е **264 – 59**, което е почти максимална стойност на unsigned long long. Това означава, че е възможен overflow при наивната имплементация.

За да се избегне това може да се подходи по следния начин.

Нека **MOD** = 264 – 59

Нека **a** и **b** са цели числа< **MOD.**

Тогава ако **MOD – a > b**, спокойно можем да намерим сумата **a + b**, тъй като тя е по-малка от **MOD**.

Ако обаче **MOD – a < b**, тогава ще намерим търсената стойност по следния начин:

**(a+b) % MOD = a + b – MOD = b – (MOD – a)**

Това е вярно, защото **a, b < MOD**, следователно **a + b < 2 \* MOD**, следователно **(a+b) – MOD < MOD**, следователно **(a+b) – MOD = (a+b) % MOD**, тъй като **MOD** е просто число.

Важното в случая, е че при пресмятането на **b – (MOD - a),** е невъзможно да получим резултат, по-голям от **MOD**, т.е. нямам как да предизвикаме overflow.