

The background of the slide features a dense arrangement of ripe pineapples. The pineapples have a characteristic yellow and brown patterned skin and long green spiky leaves. They are packed closely together, filling the frame. In the center, there's a small red sign with white text that appears to say "SALON".

# Ciąg Fibonacciego w naturze i muzyce

inż. Filip Rutkowski, inż. Kasper Seweryn

# Czym jest ciąg Fibonacciego?

Ciąg Fibonacciego jest ciągiem liczb, w którym każdy kolejny element jest sumą dwóch poprzednich elementów. Rozpoczyna on się od dwóch jedynek:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89...

Ciąg ten został odkryty przez włoskiego matematyka — Leonarda z Pizy, który uchodzi za jednego z najwybitniejszych matematyków swoich czasów.

Definicja dla informatyków :)

```
def fibonacci(n):
    if n < 2:
        return 1

    return fibonacci(n - 1) + fibonacci(n - 2)
```

# Zabawa z Fibonaccim

Co się wydarzy gdy podniesiemy liczby z ciągu Fibonacciego do drugiej potęgi?

$$1^2, 1^2, 2^2, 3^2, 5^2, 8^2, 13^2, 21^2, \dots = 1, 1, 4, 9, 25, 64, 169, 441, \dots$$

$$1 + 1 = 2 \qquad \qquad \qquad 1 + 1 + 4 = 6 \qquad \qquad \qquad = 2 * 3$$

$$1 + 4 = 5 \qquad \qquad \qquad 1 + 1 + 4 + 9 = 15 \qquad \qquad \qquad = 3 * 5$$

$$4 + 9 = 13 \qquad \qquad \qquad 1 + 1 + 4 + 9 + 25 = 40 \qquad \qquad \qquad = 5 * 8$$

$$9 + 25 = 34 \qquad \qquad \qquad 1 + 1 + 4 + 9 + 25 + 64 = 104 \qquad \qquad \qquad = 8 * 13$$

...

...

...

# Czym jest złota proporcja?

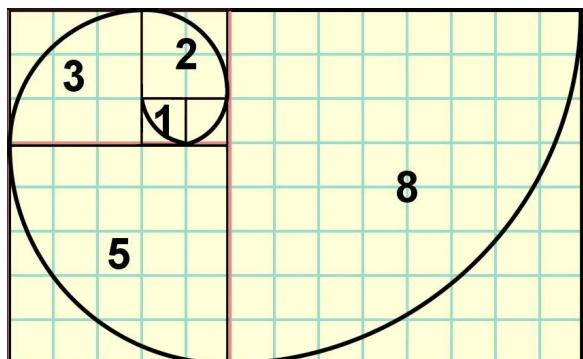
Złota proporcja jako stosunek dwóch liczb ciągu Fibonacciego znajdujących się obok siebie

Gdy zaczniemy dzielić coraz większe, sąsiadujące liczby ciągu Fibonacciego, jej wynik będzie coraz bardziej zbliżał się do:

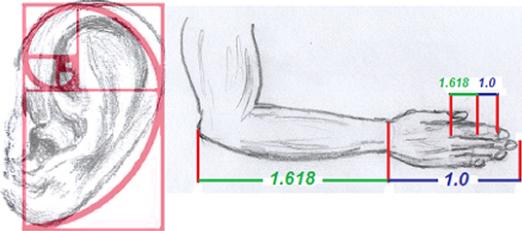
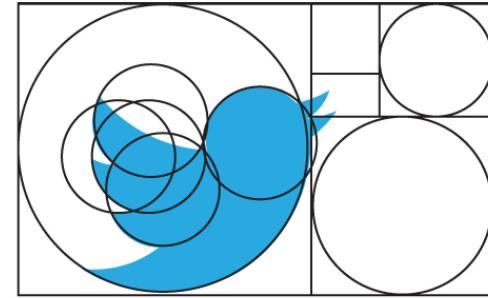
$$\phi \approx 1.618033988749\dots$$

Im większe liczby dzielimy, tym wynik jest dokładniejszy. Istotną kwestią jest to, że zarówno liczby Fibonacciego, jak i złota proporcja, występują bardzo często w otaczającym nas świecie.

Graficzną reprezentację ciągu Fibonacciego przedstawia złota spirala. Składa się ona z kwadratów o wymiarach kolejnych liczb ciągu.

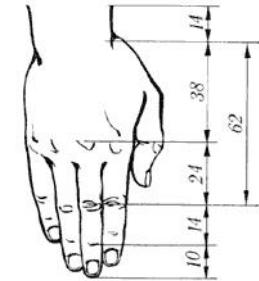
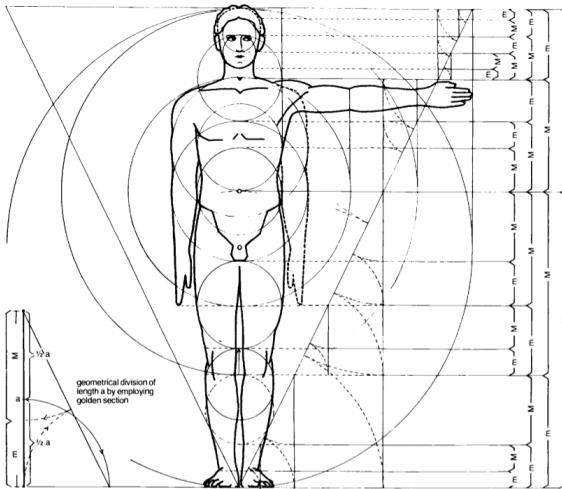
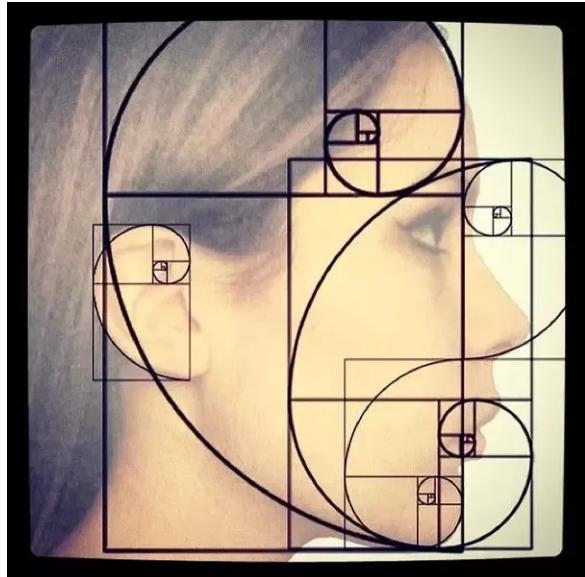


# Gdzie znajdziemy ciąg Fibonacciego oraz złotą proporcję?



# Złota proporcja w anatomii

Złotą proporcję można zauważać także w anatomii człowieka. Nie u każdego człowieka proporcje muszą być one idealnie zachowane, ale na pewno są bardzo zbliżone.



# Porozmawiajmy o ananasach!



8 parallel rows of scales spiralling gradually

13 parallel rows of scales spiralling at a medium slope

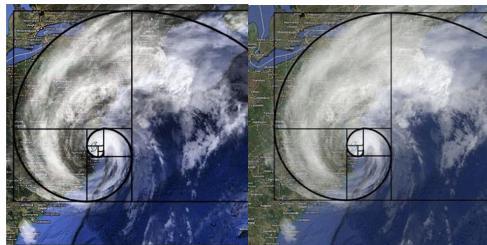
21 parallel rows of scales spiralling at a steep slope



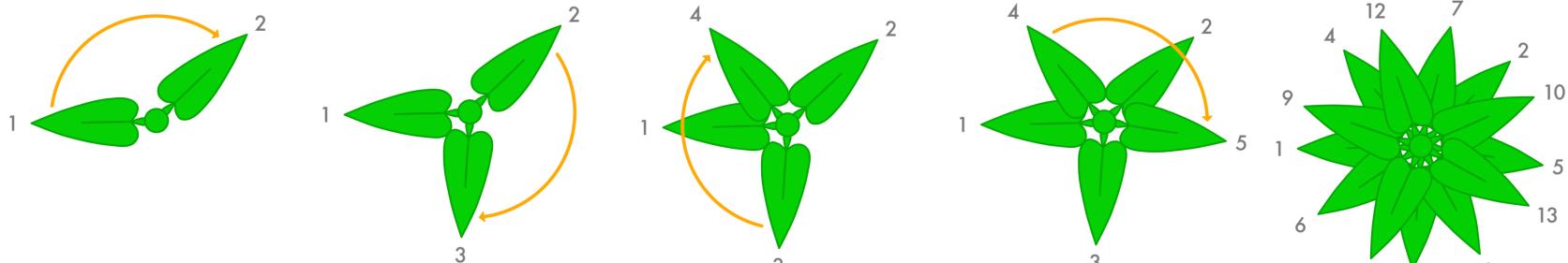
# Ciąg Fibonacciego i złota proporcja w naturze

Złoty podział występuje w:

- Kształtach spiralnych galaktyk
- Formujących się huraganach



- Ułożeniu gałęzi na pniu roślin i w nerwów liści (<https://www.nature.com/articles/s41598-018-31763-1#Fig3>)
- Ulistnieniu roślin - każdy kolejny liść rośnie co  $360(1 - \frac{1}{\phi})^\circ \approx 137.5^\circ$



# Ciąg Fibonacciego i złota proporcja w muzyce

Kolejną bardzo ciekawą informacją jest to, że ciąg Fibonacciego można stosować także w muzyce.

Kompozytorzy już dawno temu odkryli, że umiejętne wykorzystanie danych z ciągu, może ułatwić stworzenie kompozycji na bardzo wysokim poziomie. Jednym z najbardziej znanych przykładów jest tutaj Canon D-dur Johanna Pachelbela, na którym bazuje wiele znanych współczesnych utworów, jak na przykład "No Women, No cry" Boba Marleya, albo "Let It Be" zespołu "The Beatles".



# Ciąg Fibonacciego i złota proporcja w muzyce



# Ciąg Fibonacciego i złota proporcja w muzyce

TOOL — Lateralus

Black  
Then  
White are  
All I see  
In my infancy  
Red and yellow then came to be  
Reaching out to me  
Lets me see

<https://youtu.be/Y7JG63IuaWs?t=94>



# Matematyka vs Informatyka

Interaktywny przykład

```
const PHI = 1.618033988749, SQRT5_I = 1 / Math.sqrt(5)

const fib1 = (n: number) => SQRT5_I * (
  PHI ** n - (-PHI) ** (-n)
}

const fib2 = (n: number, nums = [1, 1]) => {
  if (n == 0) return 0
  if (n < 2) return 1
  for (let count = 0; count < n - 2; ++count) {
    [nums[0], nums[1]] = [nums[1], nums[0] + nums[1]]
  }
  return nums[1]
}
```

$$fib_1(1 \quad ) = 1$$

$$fib_2(1 \quad ) = 1$$

**O( $2^n$ ) Time | O(n) Space**

```
1. def fib(n):
2.   if n == 1 or n == 2: return 1
3.   return fib(n-1) + fib(n-2)
```



**O(n) Time | O(n) Space**

```
1. def fib(n, memo=[1:1, 2:1]):
2.   if n in memo: return memo[n]
3.   memo[n] = fib(n-1) + fib(n-2)
4.   return memo[n]
```



**O(n) Time | O(1) Space**

```
1. def fib(n):
2.   nums = [1,1]; count = 0
3.   if n == 1 or n == 2: return 1
4.   while count < n - 2:
5.     nums.append(sum(nums)); nums.pop(0)
6.     count += 1
7.   return nums[1]
```



**O(log(n)) Time | O(1) Space**

$$\begin{bmatrix} fib(n+1) & fib(n) \\ fib(n) & fib(n-1) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^n$$

+

MATRIX EXPONENTIATION



**O(1) Time | O(1) Space**

$$fib(n) = \frac{\phi^n - (-\phi)^{-n}}{\sqrt{5}}$$



Dziękujemy!

