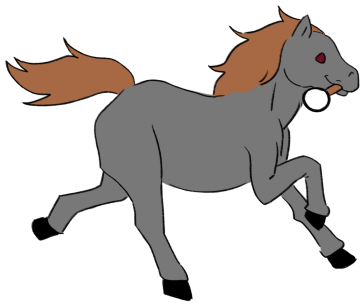
Day 1 Adventure - BestFour Landing at Greedy-ent Mart

Mission to Travel to the Past To Slow the Earth Aging	**1**
Landing at Greedy-ent Mart	**5**
Decoding the Nature: Golden Spirals, Fibonacci and Recursion	**9**
Selling CalliLens: Coin Change Challenge and Greedy Approach	**27**
Satisfied Stomach: Fractional Knapsack Problem and Greedy Approach	**34**

Mission to Travel to the Past To Slow the Earth Aging



"Look!" Dark Knight shouts as he gallops to his clan, "I found this in Grandpa's treasure box."

"Treasure box? What is it?," the BestFour soon inquire as they gather together to examine it.

Shining beneath the warmth of the sun, Dark Knight holds out a pair of glistening, immaculate glasses. However, just before the rest decide to lay a hand on it, a line of fine print catches Banana Split's eyes: *CalliLens welcomes you to the crystal world.*

"CalliLens?" Bubble Gum murmurs, "I heard of HoloLens by Microsoft on that Zero-One Land. They claim that the transparent lenses will give you an augmented reality experience. An example may be if I'm hungry, a heap of alfalfa hay will rise

in front of my nose; but when I'm thirsty, a small stream of river will run along my hooves."

"Ok Bubble Gum," Banana Split says, "moving on, we all know that *holo* can be a three-dimensional image. However, what does *Calli* mean?"

Everyone turns and looks at each other before falling silent again.

"Hmm…" Banana Split continues, "calligraphy is a style of beautiful writing, and *Calli* means beautiful in Greek...HA! I get it! We are going to see a beautiful world through these CalliLens."

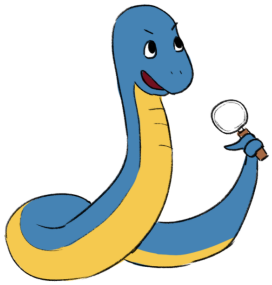
Everyone gasps.

"Really? Give it to me, " Dark Knight demands while putting on the lens, "the road is straight, and the stone is firm."

"Aish, give it to me, " Banana Split says as he snatches it away, "the bread is soft, and the cake is delicious."

"Wow! Give it to me, " Bubble Gum states as she wears on the lens, "the dream is vast, and the reality is warm."

"What is this mysterious CalliLens?" They ask as they turn to Mighty Python; the immigrant from the Zero-One Land. Mighty Python lifts his head and gazes towards the horizon before he starts explaining.



The CalliLens has a remarkable ability. When you are agitated by a monster problem, wear this gadget, and you will see that…

1) *The monster shrinks and decomposes to some miniature problems that will be easier to manage.*

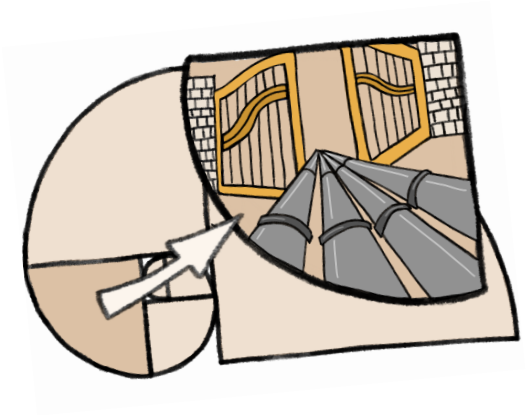
2) *The patterns or commonalities among the closely related problems are spotted, and the details that do not matter are ignored.*
3) *The step by step instruction on how to attack the monster will emerge in front of your eyes.*

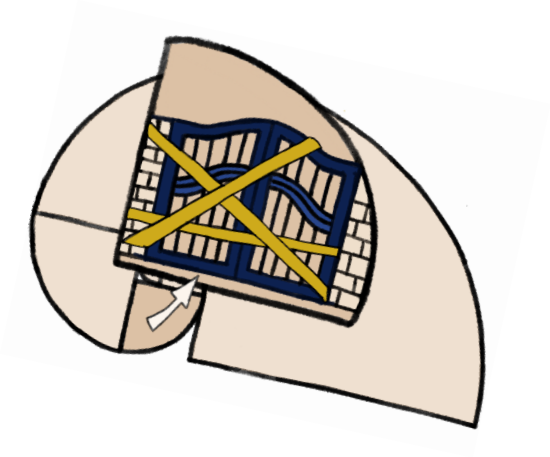
"Who is having a monster problem to attack?" Mighty Python asks while waving the CalliLens in the air.

"Me!" Banana Split hustles, "Mom asked me to clean my room today. That surely is a monster task! Where should I start?"

Bubble Gum peers at him before answering. "The CalliLens says that you should pile up your clean clothes and toss the dirty ones into the laundry bin. Also, you should place the books onto the shelf. Now this is three small tasks I can manage easily."

Mighty Python goes on, "around 100 years ago, the world was full of inventions and people started "transforming" the planet with their intelligence. However, some of the inventions were driven by people's greed to possession, greed to wealth, greed to win. They accelerated the aging process for human beings and the planet. There is only one chance to reverse the aging…(everyone is holding their breath)...





First, take the CalliLens back to the past to navigate through the chaos.

Second, seal the Gate of Traceback, the source of the toxic inventions.

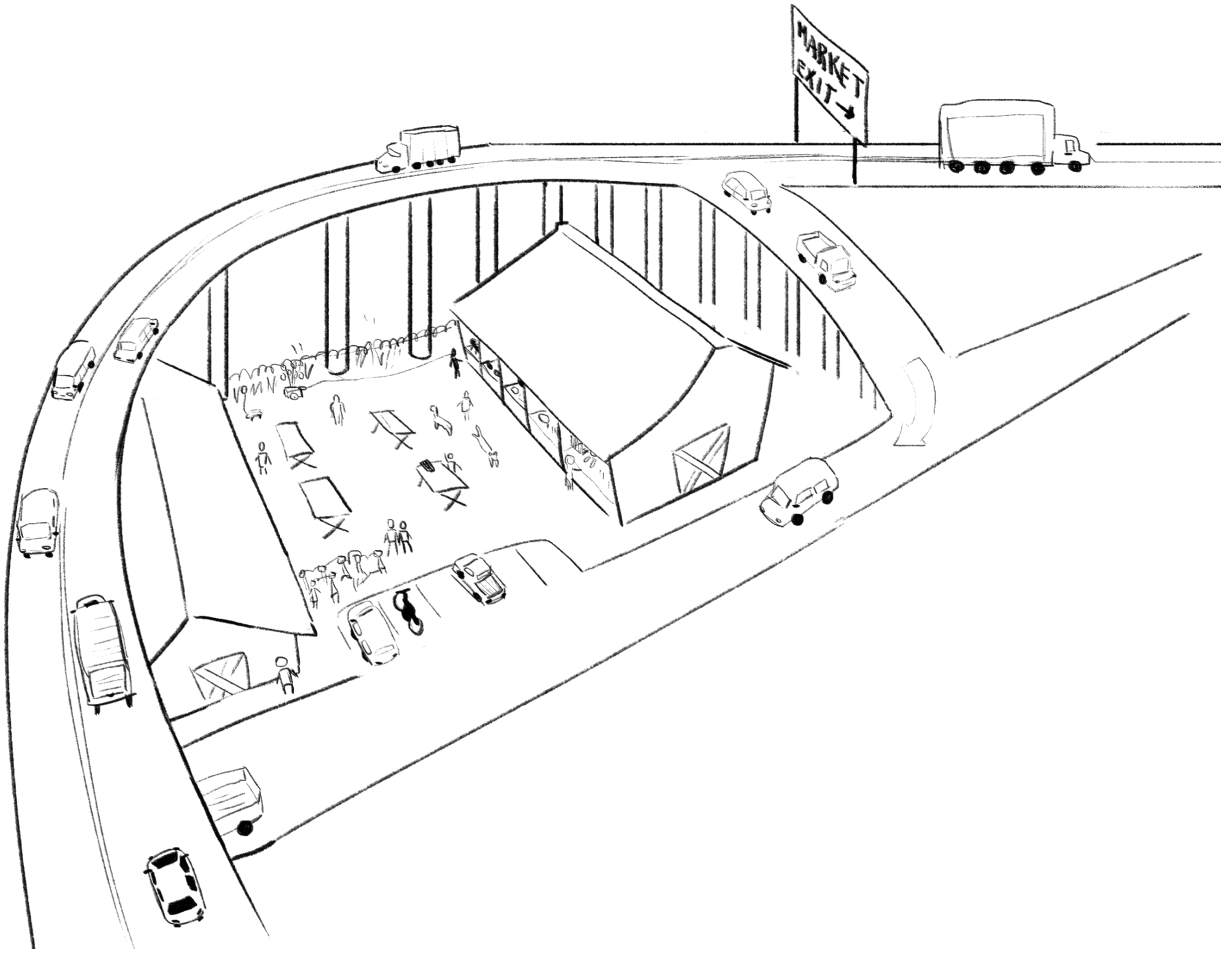
Third, build pipelines to flow the righteous inventions from the Gate of Summit to every city and town!"

"The coming week is a holiday. Let us secretly travel back and reverse the aging!" announced Dark Knight.

"Grandpa, we are going camping, just regular camping in the woods..." The BestFour waved good-bye to the family and off on the road (to be precise: time vessel).

Landing at Greedy-ent Mart

Hovering on top of a mysterious and restless land, the BestFour sees some little ants crawling along some curved lines. The lines are intertwined.



Following a hazy cloud, they landed at a market place.

There are so many mysterious and attempting foods. Banana Split reaches out for a plate of nuts.

“Wait young man, don’t you know to pay first?”, the cashier points to a box of coins and bills collected from the customers.


HONEY ROASTED PEANUTS
CHOCOLATE HAWAII MACADAMIA NUTS
Salted SUNFLOWER seeds
fresh coconut water

"Grandpa showed us once in his treasure box," Dark Knight recalls, " coins and bills are physical currency used for goods exchange."

Banana Split zooms close to the cashier's box, "well, can I pay with my fingerprint, with facial recognition, retina recognition, with any sort of biometric identification? We don't use physical currency anymore."

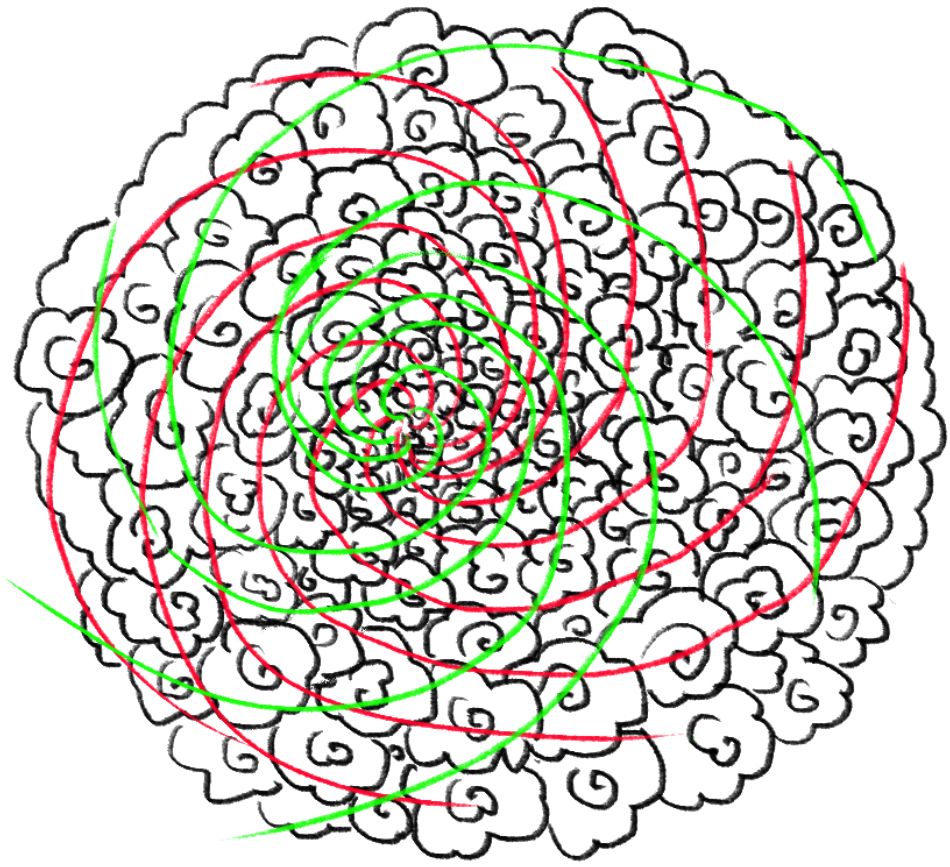
"What?! " the cashier stares at the "aliens".

“I have an idea: we can sell what we have to exchange the physical currency here.” Dark Knight announces.

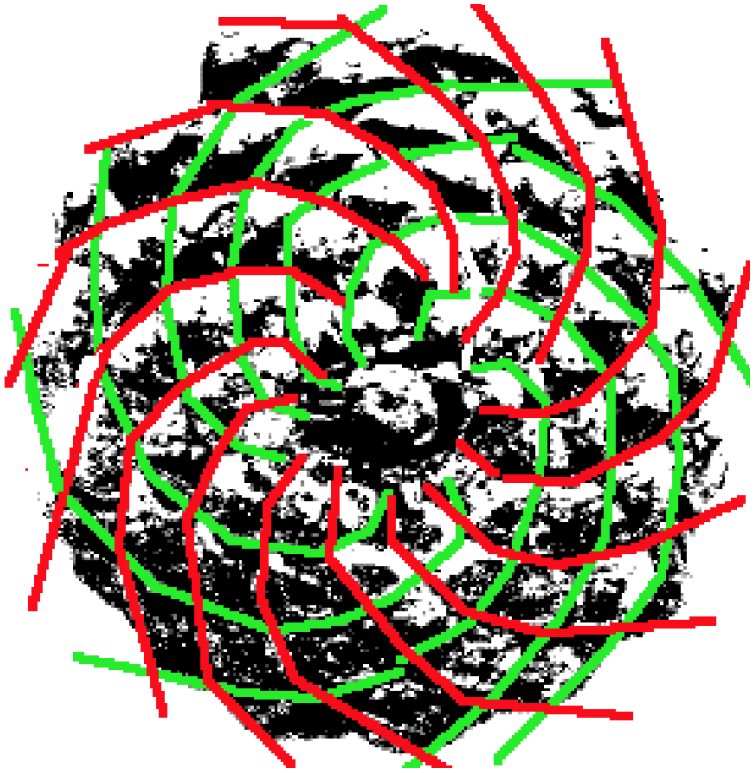
“We bought two CalliLenses. What about selling one?” Bubble Gum takes off her lens.

Decoding the Nature: Golden Spirals, Fibonacci and Recursion

A farmer comes by and tries the lens on.
“What happened to my Romanesco broccoli? I see 5 spiral lines formed by its pinnacles curling to one side, and 8 spiral lines curling to the other side. “

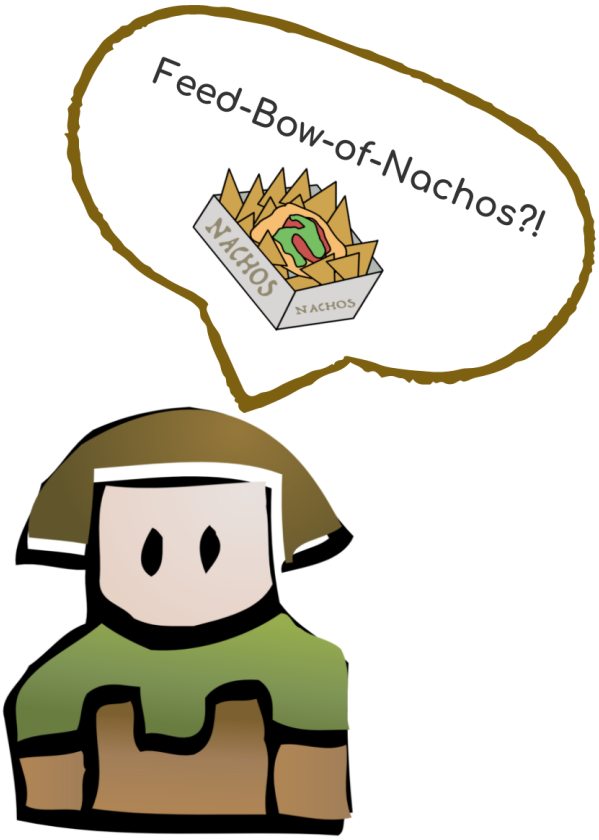


“Let me try the magic lens.” a neighbouring farmer overhears the conversation and comes by.
“OMG, this is my pine cone. It has 8 green spirals and 13 red spirals, Fibonacci number!”

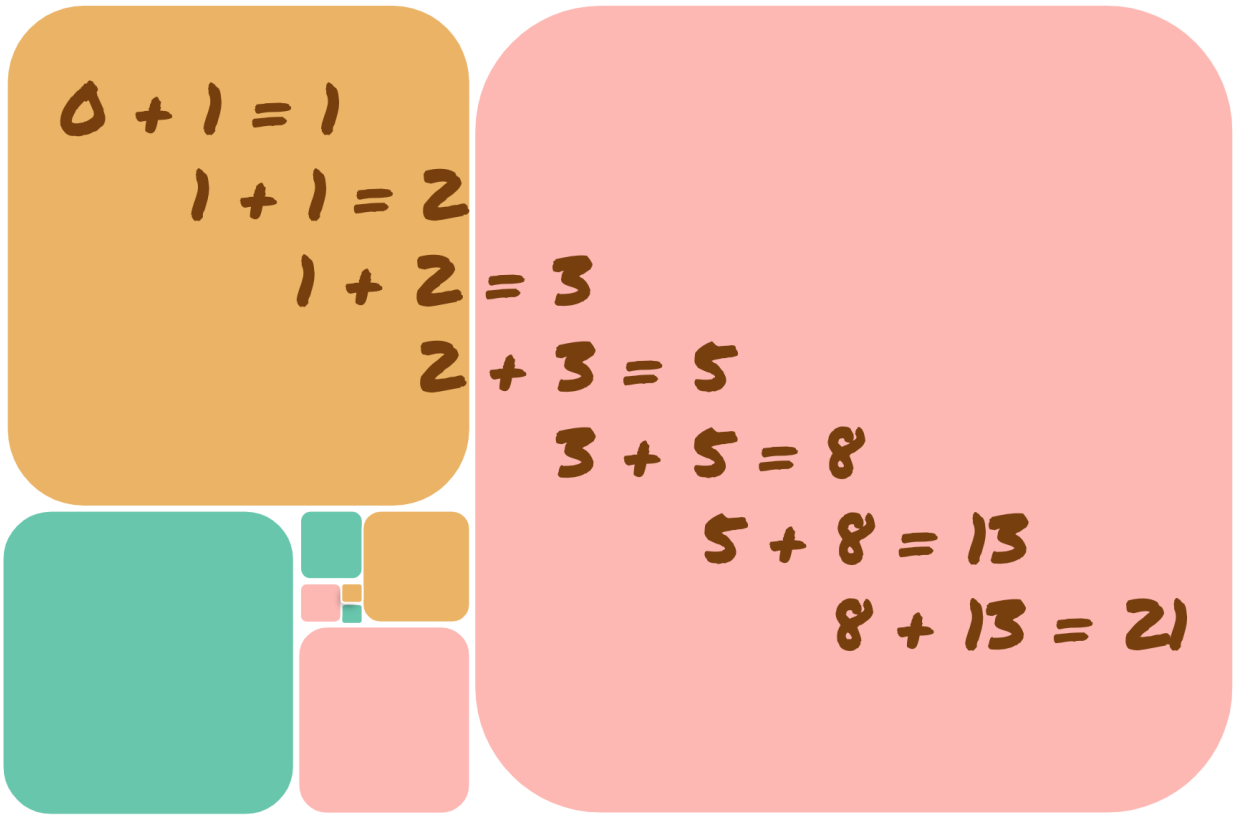


http://www.mit.edu/~levitov/FibonacciPhyllotaxis.pdf

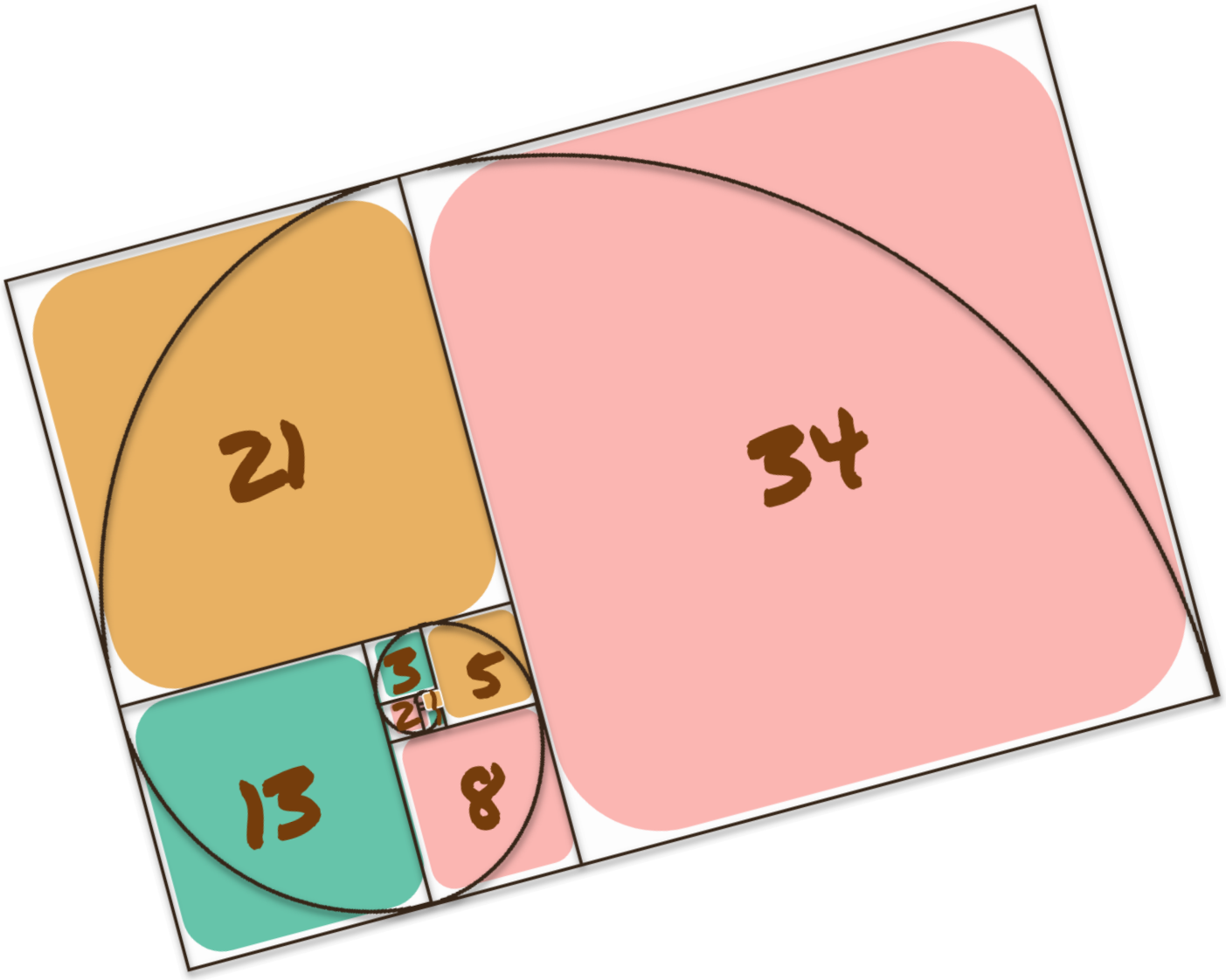
"Oh, all these spiral numbers 5, 8 and 13 are from the Fibonacci sequence." Banana Split jumped in.



“Fibonacci is a series of numbers in which each number is the sum of the two previous numbers.”



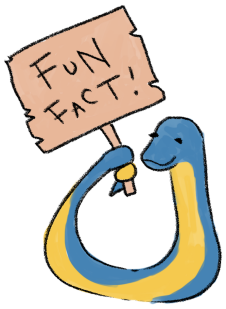
If you puzzle the squares with the Fibonacci side length together, it forms a spiral.



The ratios of Fibonacci numbers F_n/F_{n-1} approaches the factor 1.618 as n approaches infinity. This magic factor is called φ (Phi) or Golden Ratio, which was first proved by Scottish mathematician Robert Simson in 1753 (Wells 1986, p. 62). [2]

[2] Wells, D. *The Penguin Dictionary of Curious and Interesting Numbers.* Middlesex, England: Penguin Books, pp. 61-67, 1986.

Therefore, this special spiral is called Golden Spiral. It gets wider (i.e. further from its origin) by a factor of 1.618 for every quarter turn it makes.



Reality Check

Fibonacci Numbers? Leonardo Fibonacci, the man responsible for re-discovering the amazing Fibonacci Numbers, was an Italian mathematician who was born somewhere around 1170. In 1202 he published a book called 'Liber Abaci' (Book of Calculation).

He was also the man who gave us our decimal numbering system. He figured out that it was far more efficient than the Roman Numeral System, which did not even contain a '0' - zero.

https://www.all-my-favourite-flower-names.com/fibonacci-numbers.html

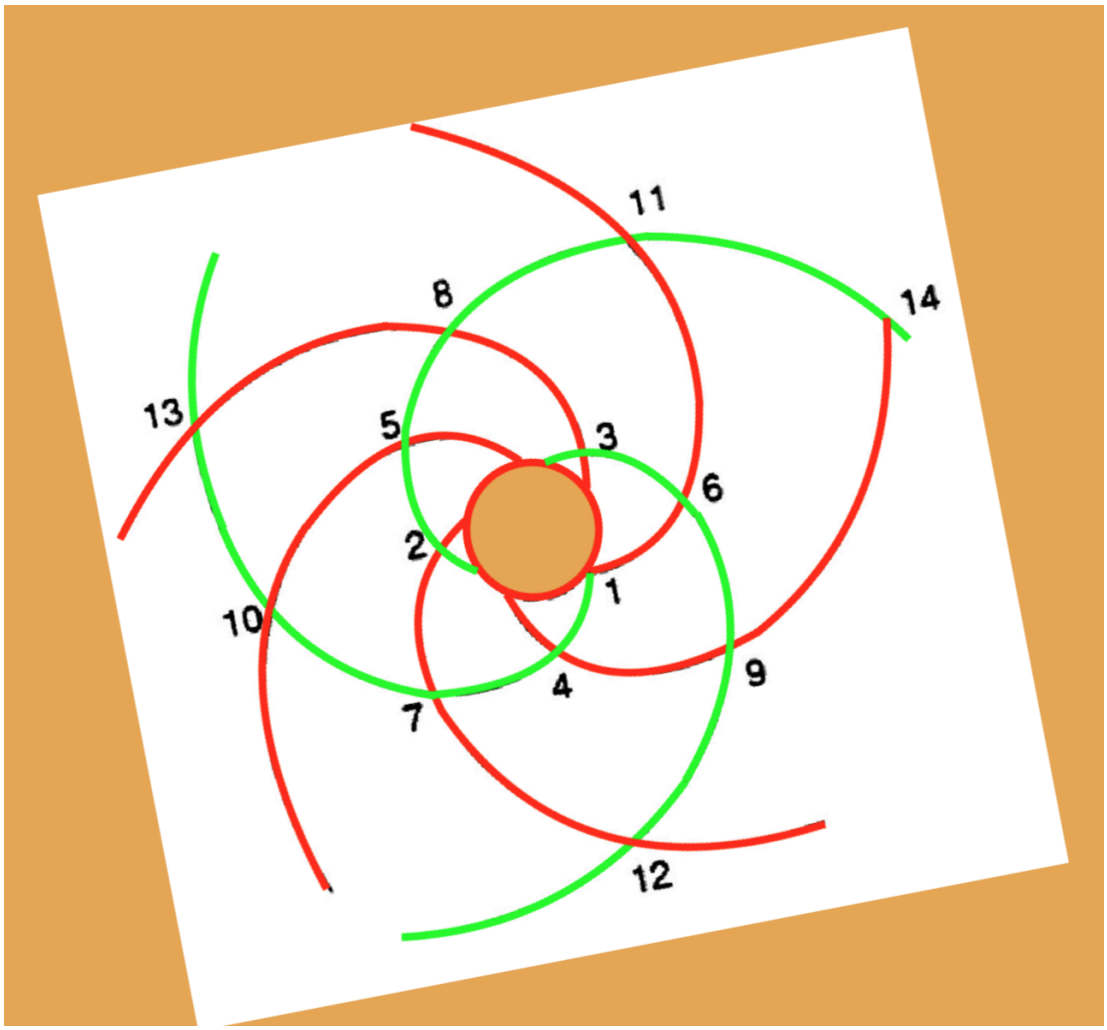
(OEIS A000045)

Besides Finobanni, there are Lucas numbers which are also special: 1, 3, 4, 7, 11, 18, 29, 47, 76…

Statistics for cones of pine-trees (Norway):
95% Fibonacci
4% Lucas
1% deficient
http://www.mit.edu/~levitov/FibonacciPhyllotaxis.pdf

Bubble Gum adds, "Fibonacci numbers are very common in nature."

"Let me try." the neighbouring gardener is tuned into this conversation.
"OMG, this is my flower."

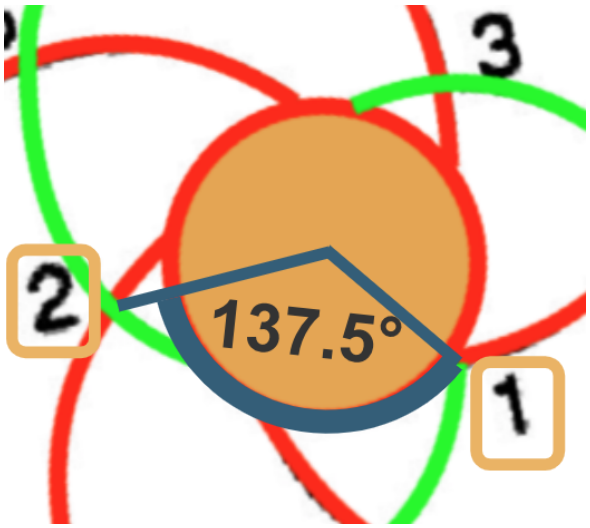


Smaller numbers represent younger petals.

3 green clockwise spirals and 5 counter-clockwise red spirals. Again, 3 and 5 are Fibonacci numbers! And the spirals are golden spirals!

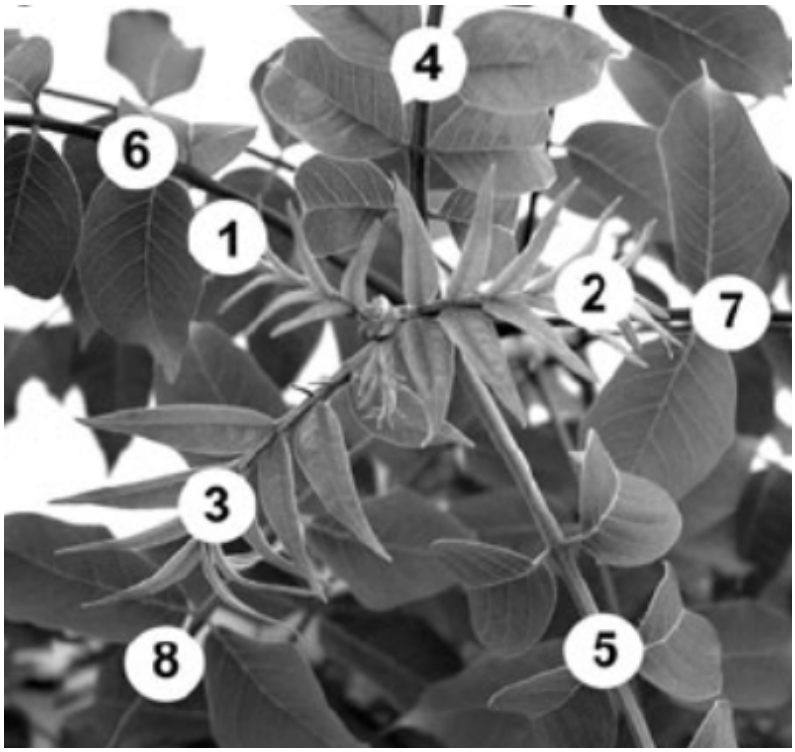
http://www.mit.edu/~levitov/FibonacciPhyllotaxis.pdf

The adjacent petals are always apart by a constant angle, 137.5°!



137.5° between petal 1 and 2, same 137.5° between petal 2 and 3, and so forth.

The crowd has formed around the BestFour by now.
"Look at the branches on the tree!" someone shouted.



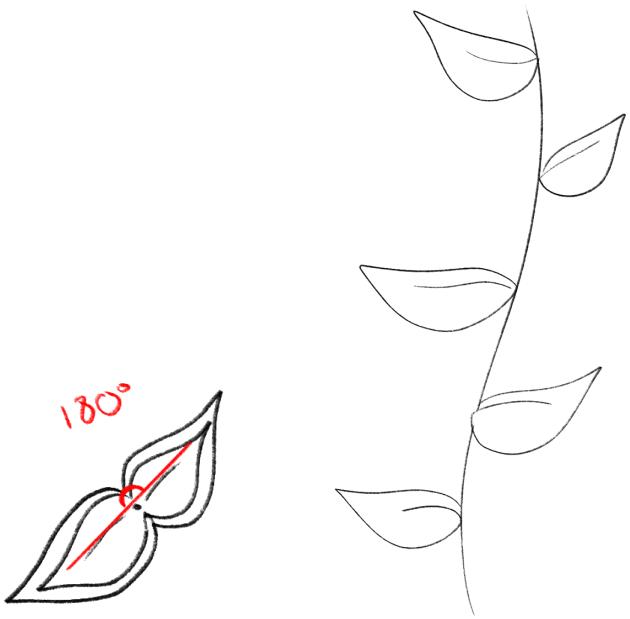
Branches of one Ailanthus altissima tree. Smaller numbers represent younger branches

https://www.researchgate.net/publication/7663446_Regulation_of_phyllotaxis
The branches follow the same 137.5° angle to diverge from each other.

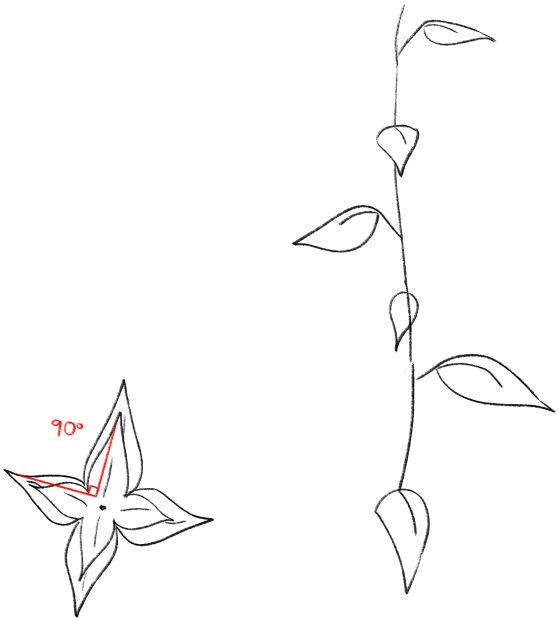
You might ask: why do plants follow this magic angle and hence form Fibonacci number of spirals (3, 5, 8, 13, etc.) for their growth? In fact, this was the question raised many times in history.

To decode the nature's mystery, let us experiment some hypotheses:
What if the leaves grow at the same angle?

Or at a 180 degree angle?



Or 90 degree angle?



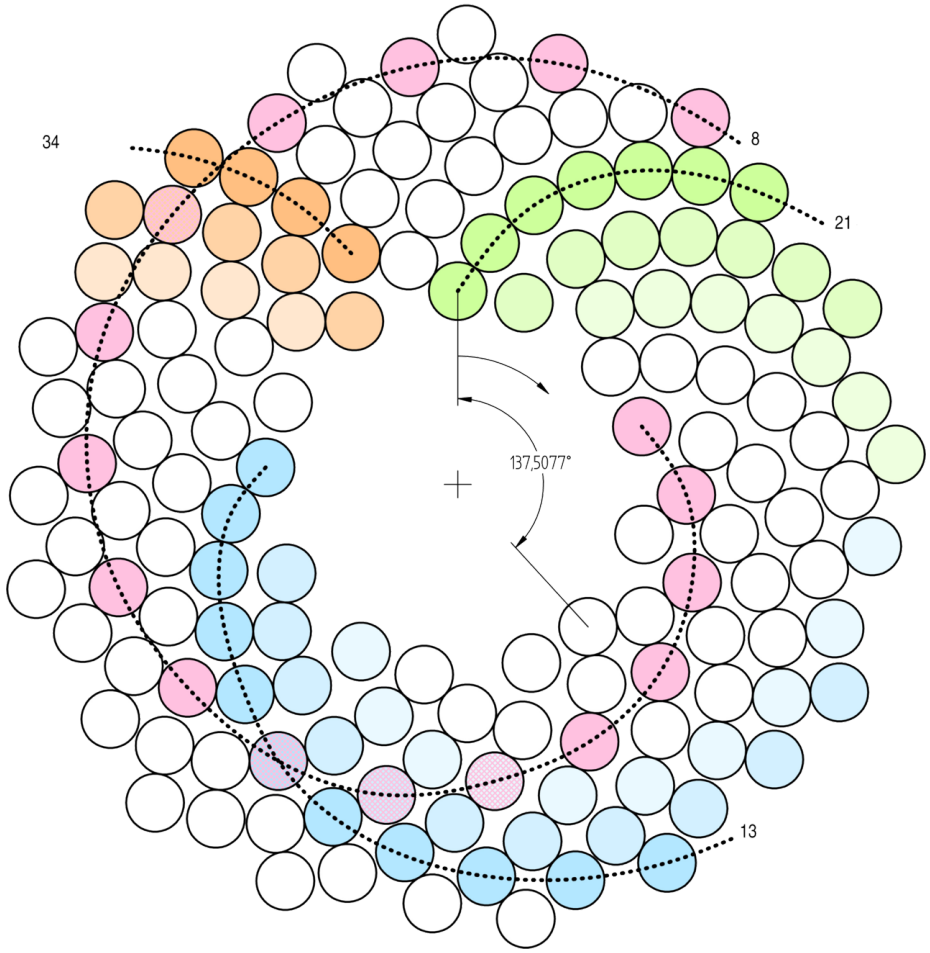
60? 30? 10? 360/n degrees?

Eventually, leaves will repeat the same cycle and overlap.

Is there a magic angle so that leaves will never overlap and therefore get maximum sunlight?

Golden angle Ψ ≈ 137.5077°

That is the exact angle we observed on pine cones, flowers and tree branches!

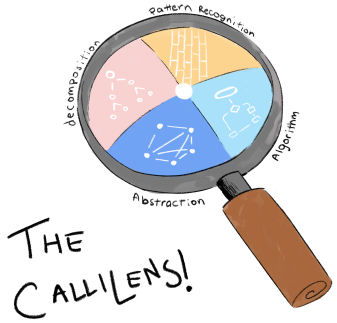


Mathematical Abstraction

Arrangement of equal-sized circles at a distance of the golden angle Ψ ≈ 137.5077° with colored Fibonacci spirals

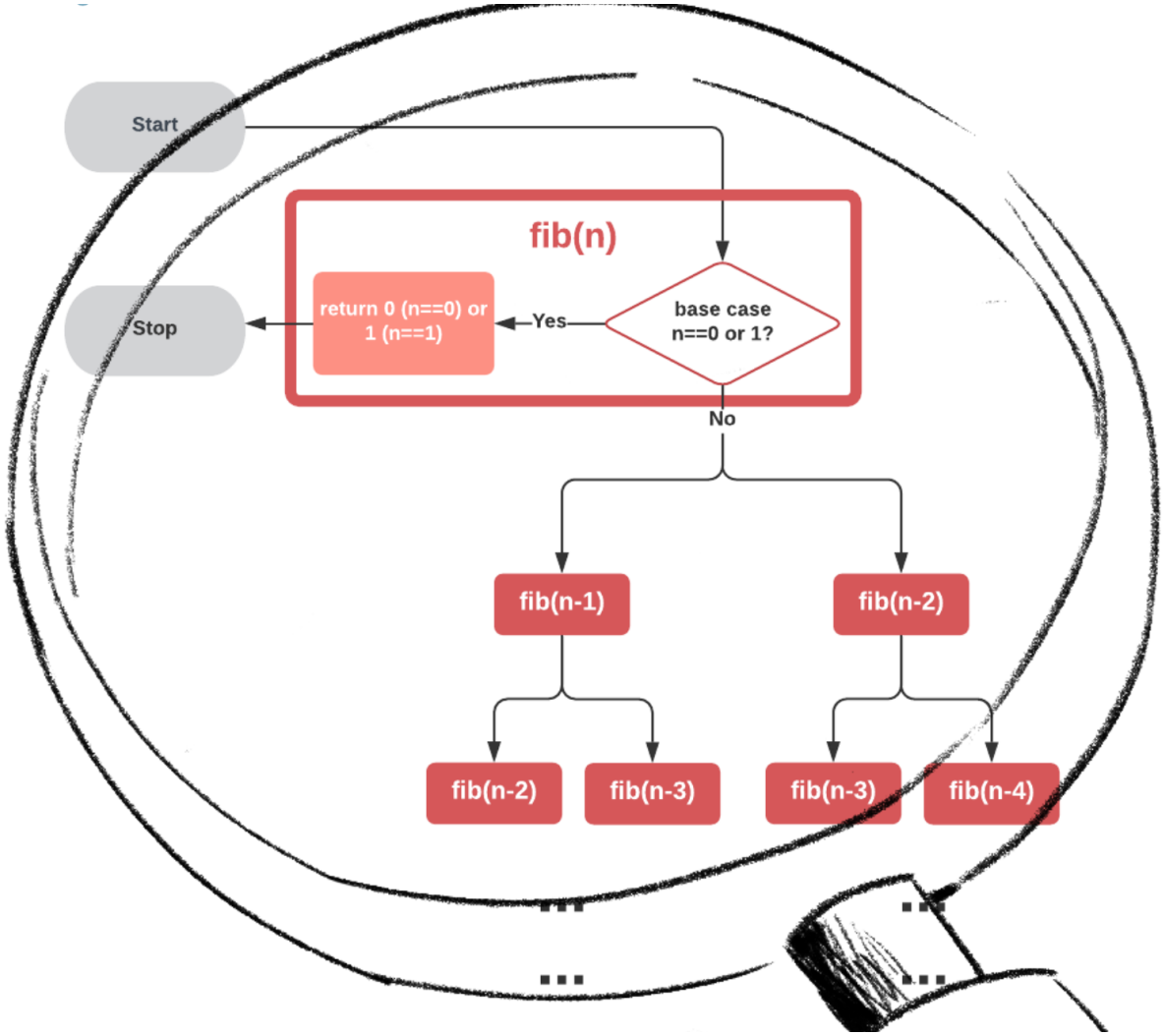
https://commons.wikimedia.org/wiki/File%3AFibonacci_numbers.jpg

Ha, nature loves Fibonacci and the Golden Angle 137.5°!



"Then, how to compute the Fibonacci number sequence?" one of the farmers asks. Bubble Gum points to the lens, "it will tell…"
Wow…

- Say function fib(n) is used to compute the nth Fibonacci number.
- Per Fibonacci definition, the function calls itself to add the previous two numbers together fib(n-1) + fib(n-2).
- When we reach the first two numbers in the Fibonacci sequence (0 and 1), the function will stop the calling chain and return 0 and 1 respectively.

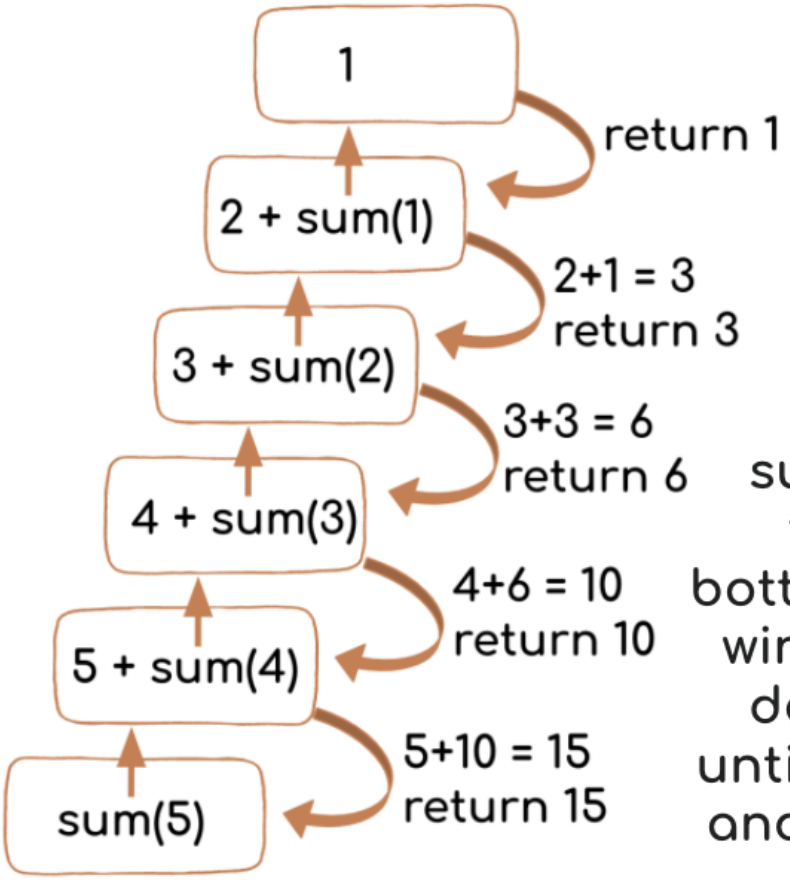


A function calls itself! Right, such a function is called recursive function.

Recursion

A function calls itself and stops the chain at base case.

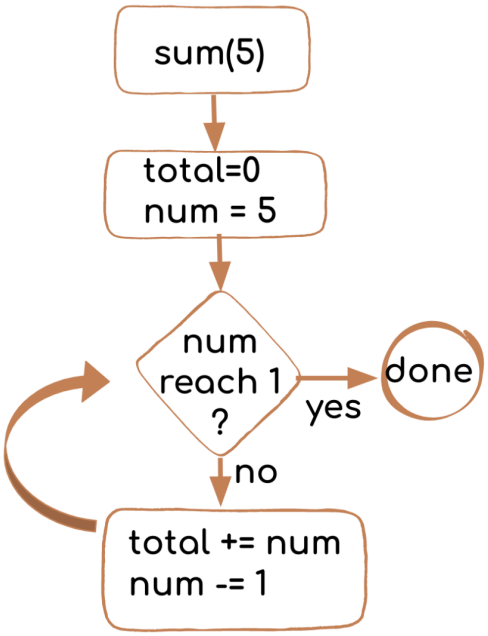
Recursive Approach



sum() starts from the bottom of stack, winding up to decompose until base case, and unwinding down to compute the result.

As a comparison, let us present the regular iterative approach using a loop below.

Iterative Approach

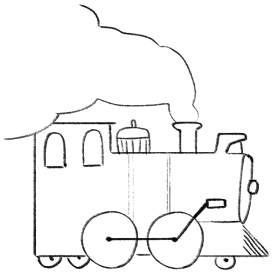


Reality Check

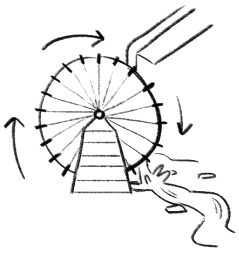
The first person brought up the "definition by recursion" is Dedekind in his essay *Was sind und was sollen die Zahlen* (1888). He recursively defined addition, multiplication, and exponentiation [10]

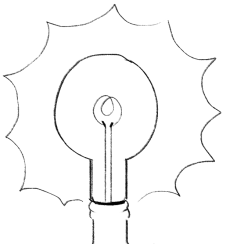
[10] https://plato.stanford.edu/entries/recursive-functions/

Think about human beings advancing technologies generation after generation.

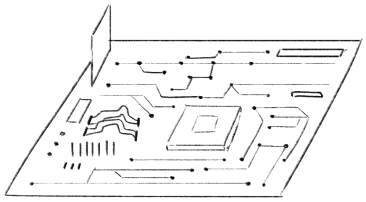


Steam machines and water power in the 1st industrial revolution

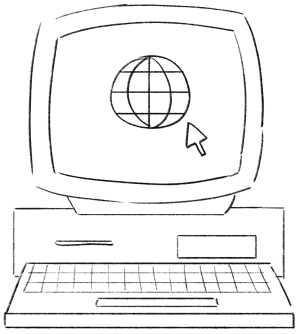




Electricity in the 2nd Industrial Revolution

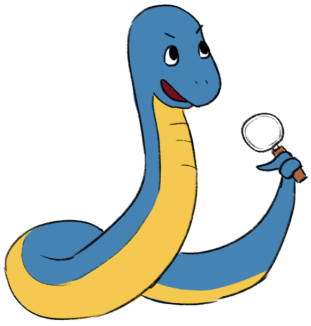


Computing in the 3rd Industrial Revolution



Current artificial intelligence and robotics in the 4th industrial revolution.

Where will this trend lead human beings? Time will tell. But one thing for sure is that the toxic inventions will stop soon! Right, that is our mission!



Now let us look at the Python solution using an iterative and recursive approach respectively.

Iterative Approach

```python
# Apple Pi Inc.
# Algorithmic Thinking
# Fibonacci Numbers
# Iterative Approach

# iteration while loop
def fibonacci(n):
    # two initial numbers
    # in Fibonacci sequence
    a = 0
    b = 1

    i = 0
    while i < n:
        next_num = a + b
        a = b
        b = next_num
        i += 1

    return a

for i in range(10):
    print(fibonacci(i), end=" ")
```

Recursive Approach

```
# Apple Pi Inc.
# Algorithmic Thinking
# Fibonacci Numbers
# Recursive Approach

# recursive function
def fibonacci(n):
    # base case to stop the recursion call
    if (n==0):
        return 0
    if (n==1):
        return 1

    # compute fib number
    return fibonacci(n-1) + fibonacci(n-2)


for i in range(10):
    print(fibonacci(i), end=" ")
```

Introduction to Pygame

Selling CalliLens: Coin Change Challenge and Greedy Approach

By this time, CalliLens is getting so popular at Greed-ient Mart. We can sell it for a good deal so that we have cash to buy food and other stuff!

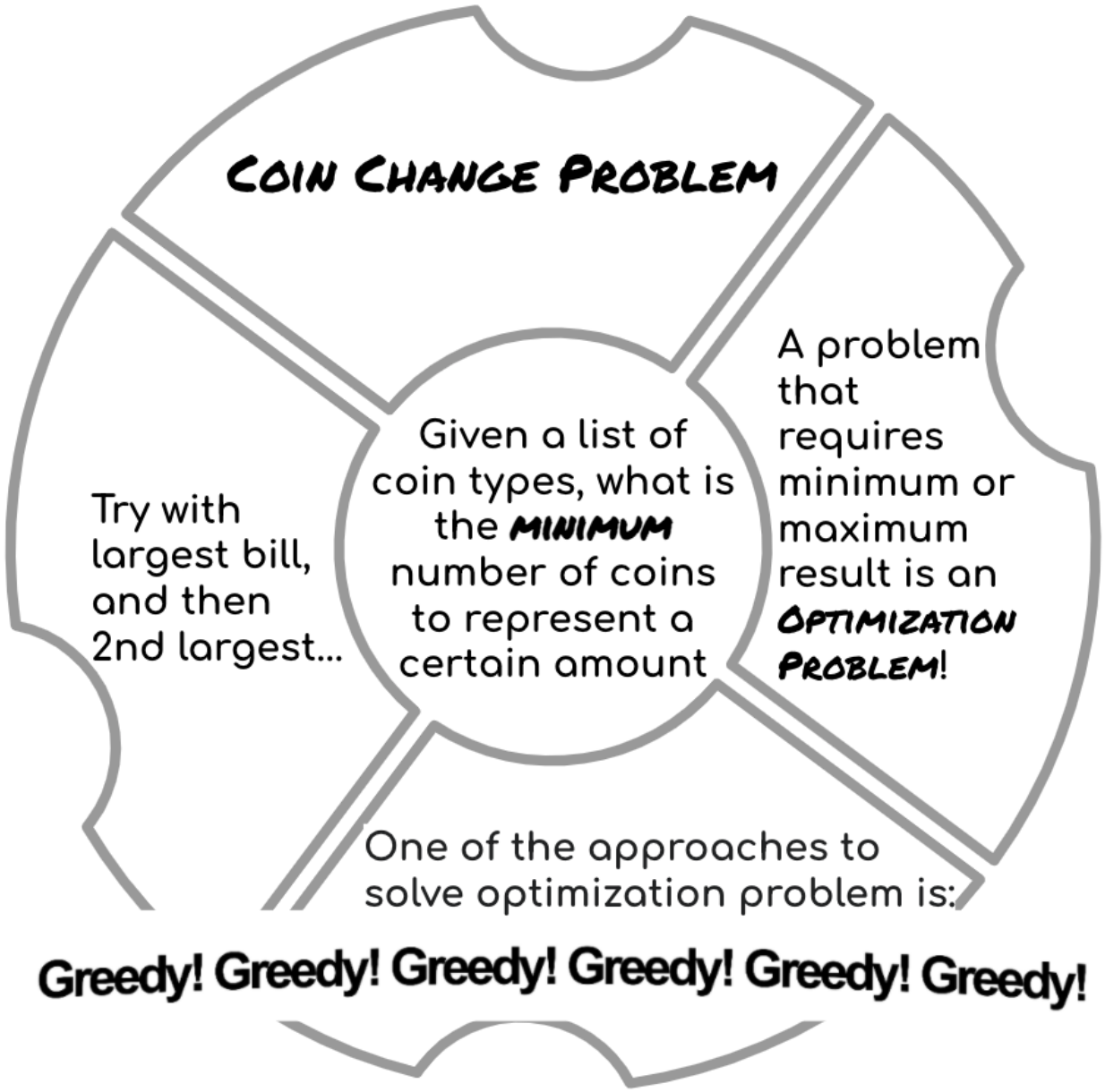
"$100", "$200", "300", "320" gold sparkles start jumping out of the eyes of the BestFour.
"Ok, ok, why don't we take the average of the four prices and sell the lens for $230?" Dark Knight makes the final call.

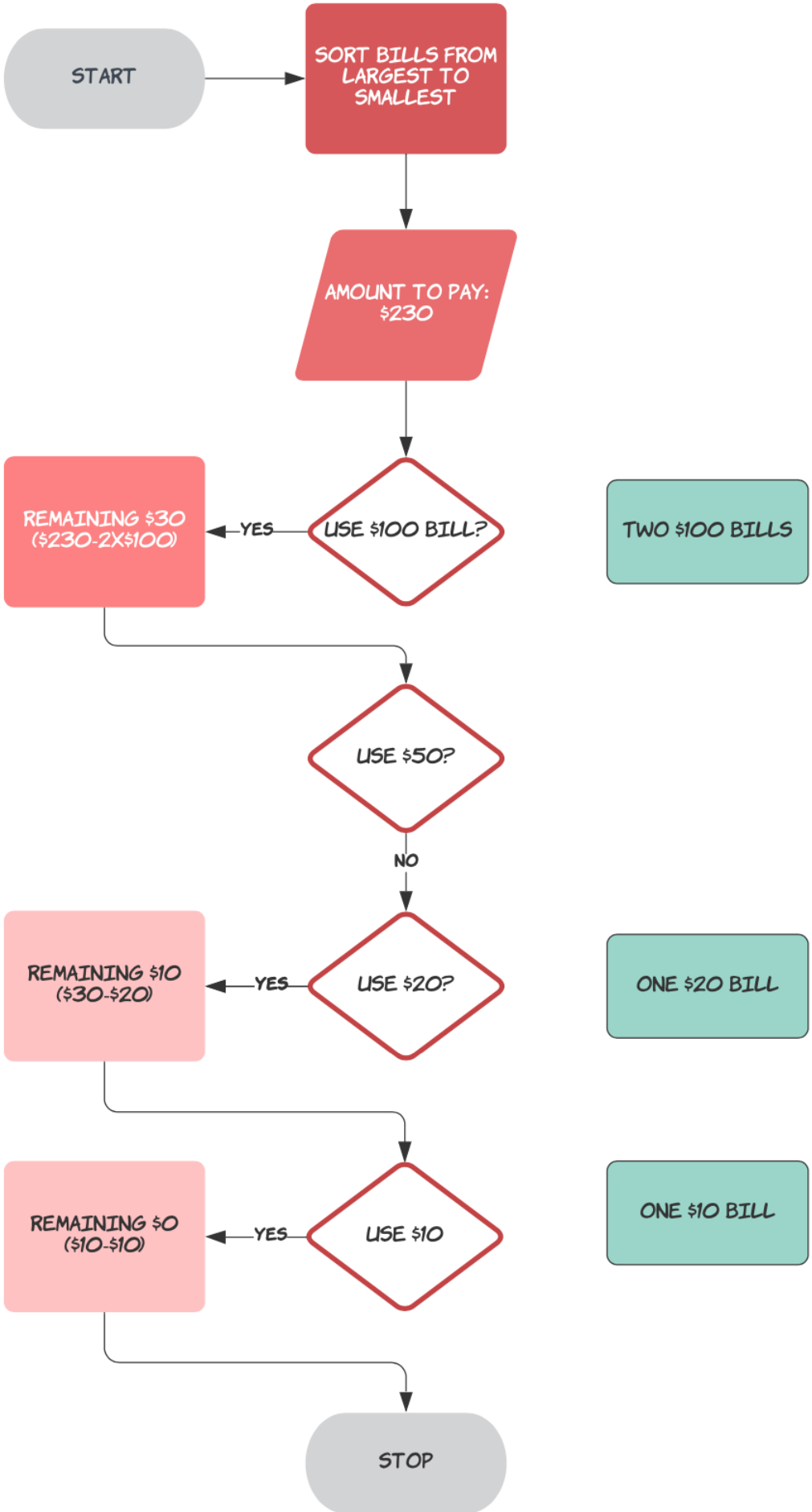
The farmer in front of the line immediately takes out a big sack of money and starts piling up the pennies.

"Stop, stop", Bubble Gum hurries, "it will be too heavy for us to carry. Please give us the least number of bills and coins!"

"Well…" the farmer is shuffling the paper bills and coins back and forward in his hands, "this way, oh, that way is better… and yet a better way…"

"Now you are empowered with the CalliLens. Why don't you look through it?" Banana Split hints him.


Coin Change Problem
Given a list of coin types, what is the MINIMUM number of coins to represent a certain amount
A problem that requires minimum or maximum result is an Optimization Problem!
Try with largest bill, and then 2nd largest...
One of the approaches to solve optimization problem is:
Greedy! Greedy! Greedy! Greedy! Greedy! Greedy!

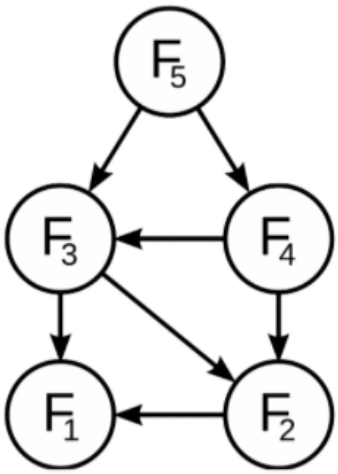

START
SORT BILLS FROM LARGEST TO SMALLEST
AMOUNT TO PAY: $230
USE $100 BILL?
YES
REMAINING $30 ($230-2X$100)
TWO $100 BILLS
USE $50?
NO
USE $20?
YES
REMAINING $10 ($30-$20)
ONE $20 BILL
USE $10
YES
REMAINING $0 ($10-$10)
ONE $10 BILL
STOP

Greedy approach

One of the approaches to solve optimization problems. Greedy works for the problems with the following two properties:

Optimal sub-structures

The optimal solution to the local context (i.e. sub-problems) is part of the optimal solution to the overall problem.

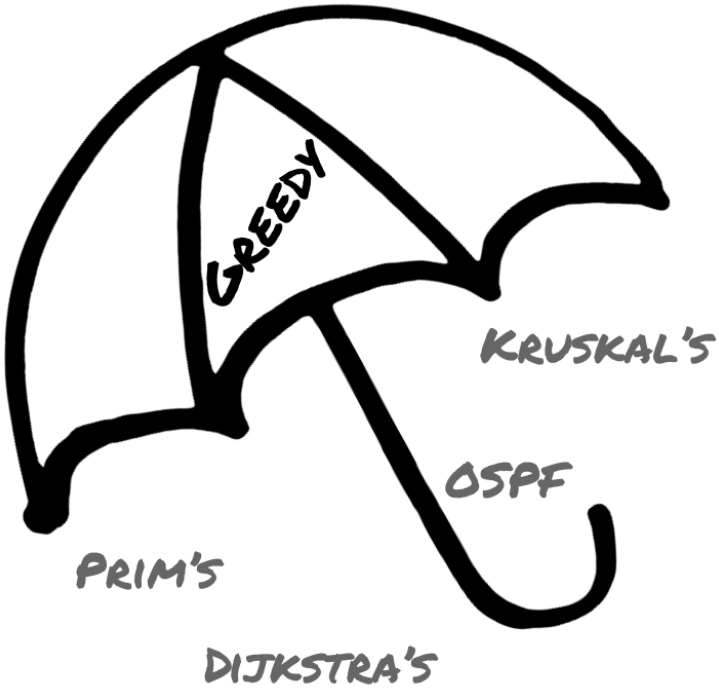


Greedy property

Greedy is 'short sighted'. It makes optimal choice at each step based on the local context with the hope of eventually reaching the globally optimal solution. It never reconsiders the previous choices.

Whereas if the optimization problem does not have above properties, we need to consider other approaches such as Dynamic Programming (DP) (next chapter in the book).

Greedy is an umbrella term for many algorithms.



- Algorithms to walk through graphs, like Prim's and Kruskal's algorithms to minimize path costs across multiple locations (Day 3 Adventure in the book).

- Dijkstra's algorithm to find the shortest path from a given source location (Day 4 Adventure in the book).

- Network protocols such as the open-shortest-path-first (OSPF) and other network packet switching protocols to minimize time spent on a network.

Satisfied Stomach: Fractional Knapsack Problem and Greedy Approach

"Now we have enough cash from selling CalliLens. I can't wait for the food quest to satisfy my empty stomach!" Banana Split is about to run to the food stands.

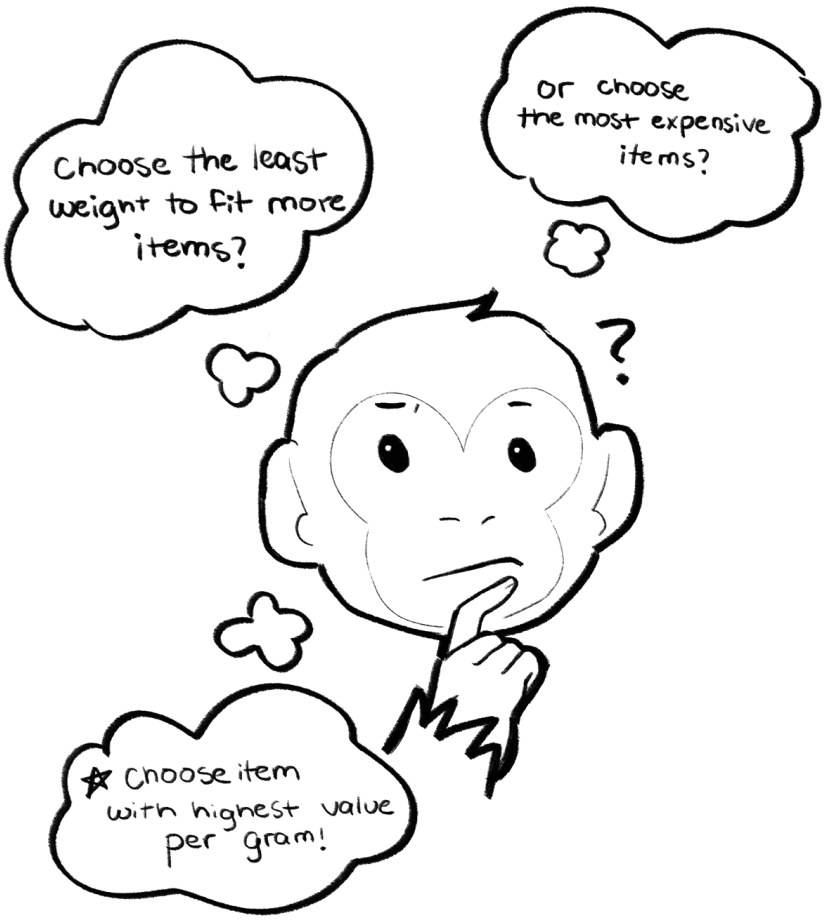
"Remember, you have limited stomach size, you got to be careful about your choices?" Dark Knight chases him behind.

"How can I get the highest satisfaction with my stomach capacity?" Bubble Gum asks himself.

Stomach Capacity: 100g
Can take fractional up to 1 serving per food type

Food Type	Honey Roasted Peanuts	Salted Sunflower Seed	Coconut Water	Chocolate Hawaii Macadamia Nuts
Satisfaction Value	90	45	140	90
Grams Per Serving	30g	30g	80g	20g

Banana Split can not fit in all his favorite nuts in his stomach, he will have to choose to maximize the satisfaction value. Maximization problem is an optimization problem. We will think of Greedy approach.



What is your idea to help Banana Split?

What? See through CalliLens and it will tell. Did I hear you right? Smart!

The abstraction of this "Banana Split filling his stomach" problem is a typical Fractional Knapsack problem: given a set of items with specific value and weights, the goal is to maximize the value in a knapsack (in this case total satisfaction value) within the weight constraint (Banana Split's stomach).

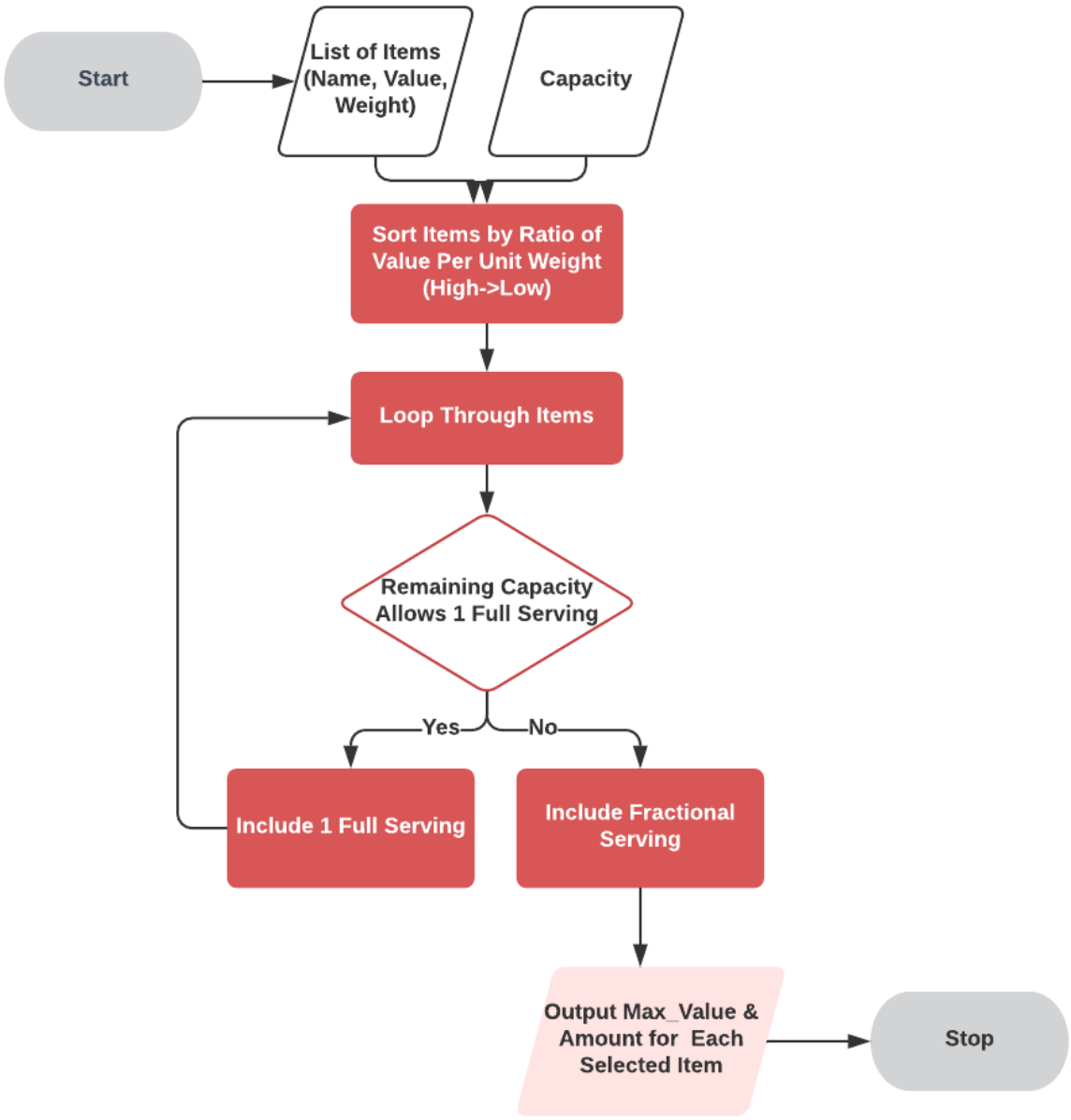
Since Banana Split can eat fractional serving (e.g. half a serving of Salted Sunflower Seed), this type of Knapsack problem is called Fractional Knapsack Problem (or Continuous Knapsack Problem).

Greedy Solution Steps:

1. Sort items by value per unit weight (i.e. ratio of value/weight) from high to low
2. Greedily take high ratio items as long as it fits capacity
3. If can't take full serving, take fractional serving
4. Done!

Food Type	Honey Roasted Peanuts	Salted Sunflower Seed	Coconut Water	Chocolate Hawaii Macadamia Nuts
Satisfaction Value	90	45	140	90
Grams Per Serving	30g	30g	80g	20g
Satisfaction Per Gram	3	1.5	1.75	4.5

Diagram Illustration



Mighty Python

factional_knapsack_greedy.py at github...

```python
# Apple Pi Inc.
# Algorithmic Thinking
# Fractional Knapsack Problem
# Greedy Approach

class Item:
    def __init__(self, name: str, value: int, weight: int):
        self.name = name
        self.value = value
        self.weight = weight

```

```python
# return (max_value, fractions)
#   max_value: the maximum value of selected items within capacity
#   fractions: dictionary showing selected items and the serving fractions (0,1]
# argument items: choices
# argument capacity: the maximum weight
def fractional_knapsack(items, capacity):

    # reverse sort the items by value/weight ratio (high ratio on top)
    # lambda represent a small function without name (called anonymous function)
    # format 'lambda arguments: expression'
    # here, the argument is each item in the items list
    # the expression is the ratio
    # use this lambda function to sort items by ratio in reverse order
    items.sort(key=lambda item: item.value/item.weight, reverse=True)

    max_value = 0
    fractional_amt = {}

    # Greedy approach to take high value/weight ratio items by order
    for item in items:

        if item.weight <= capacity:
            # include the whole serving
            fractional_amt[item] = 1
            max_value += item.value
            capacity -= item.weight
        else:
            # include partial serving
            fraction_to_add = capacity/item.weight
            fractional_amt[item] = fraction_to_add
            max_value += item.value * fraction_to_add
            break

    return max_value, fractional_amt

```

```
# initialize item list and capacity
items = [Item('Honey Roasted Peanuts', 90, 30),
         Item('Salted Sunflower Seed', 45, 30),
         Item('Coconut Water', 140, 80),
         Item('Chocolate Hawaii Macadamia Nuts', 90, 20)]
capacity = 100

# call knapsack function
max_value, fractions = fractional_knapsack(items, capacity)

# output result
print('The maximum value of items that can be taken within capacity:', max_value)
for item in fractions:
    print("item: " + str(item.name) + ', ', end='')
    print("amount: " + str(fractions[item]))
```

Bingo! Stomach is happy, Banana Split is happy.

It is getting dark. The night is approaching. The BestFour settles in the barn at the Greedy-ent Mart.

Bird chirping...
"Hurry, wait up! Today we need to find the map to locate the Gate of Summit and the Gate of Traceback!"

"Sir, do you know where most kinds of maps are stored?" Dark Knight asks a favor from the owner of the nearby newspaper stand.

"Centermount Academy." he points in that direction.

"Vroom..." the Vessel is flashing by the coconut stand, the peanut stand, the Hawaii nut stand, the xxx stand, yy stand, z stand...

"You're going to get a traffic speeding ticket!!!" the x stand owner is shouting,
yy stand owner: for sure
zzz stand owner: certainly