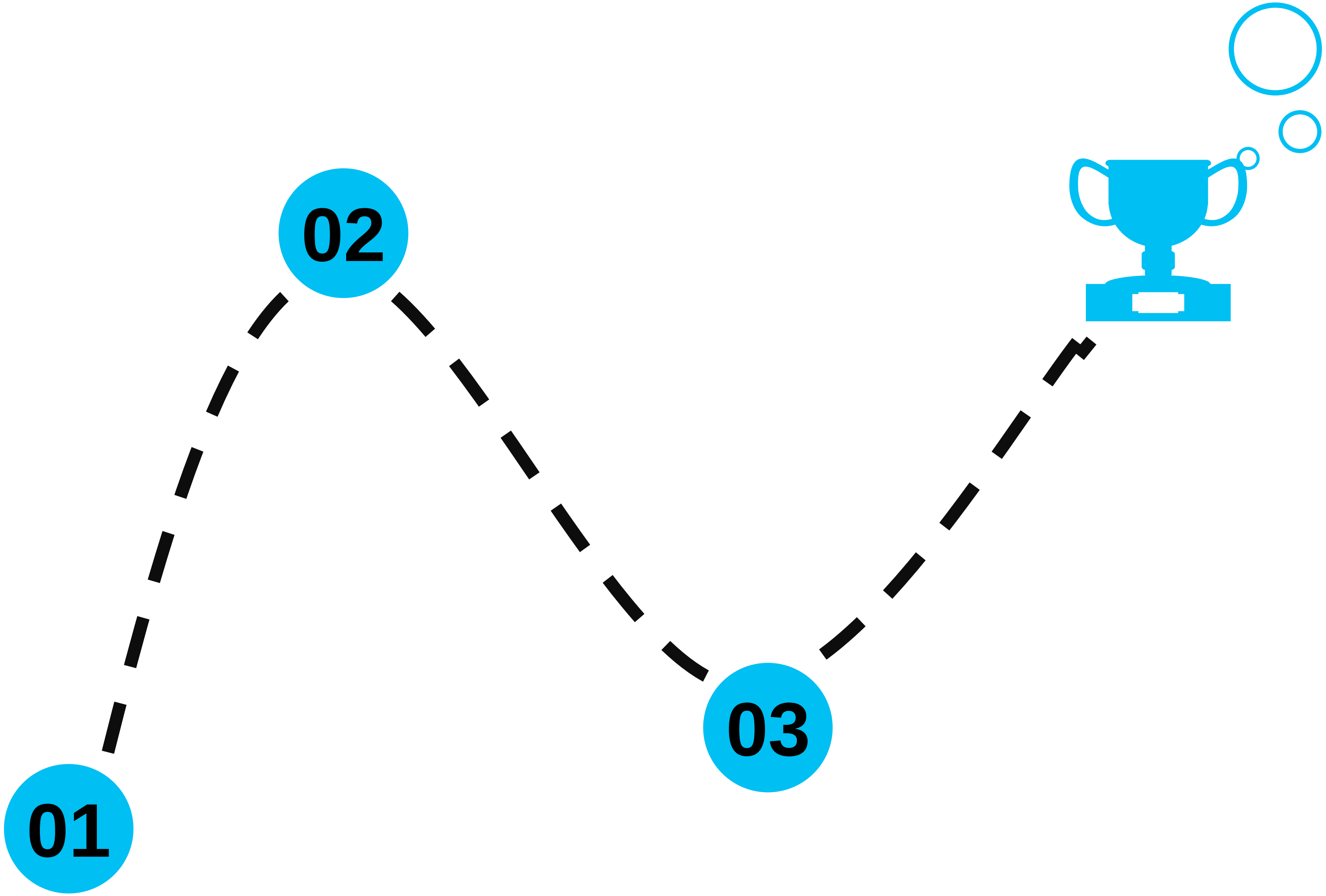
Practice: Algorithm Design

Group 13:
Nguyễn Thị Thảo Hiền
Nguyễn Thị Ly
Đặng Xuân Mai

LET’S FIND OUR REWARD!!!



Problem 1

Going Home

Abstraction

Patter Regcontion

Decomposition

Algorithm

Problem 1

Going Home

Abstraction

Find 4 indices – 2 pairs of numbers which have same sum with each other.

Problem 1

Going Home

$O(n^4)$

Algorithm

Brute Force

```
for.............
        for...........
                for............
                        for..........
```

Problem 1

Going Home

It’s something that store results of previous pairs, then compare!!!

Problem 1

Going Home

Patter Regcontion

Hash table

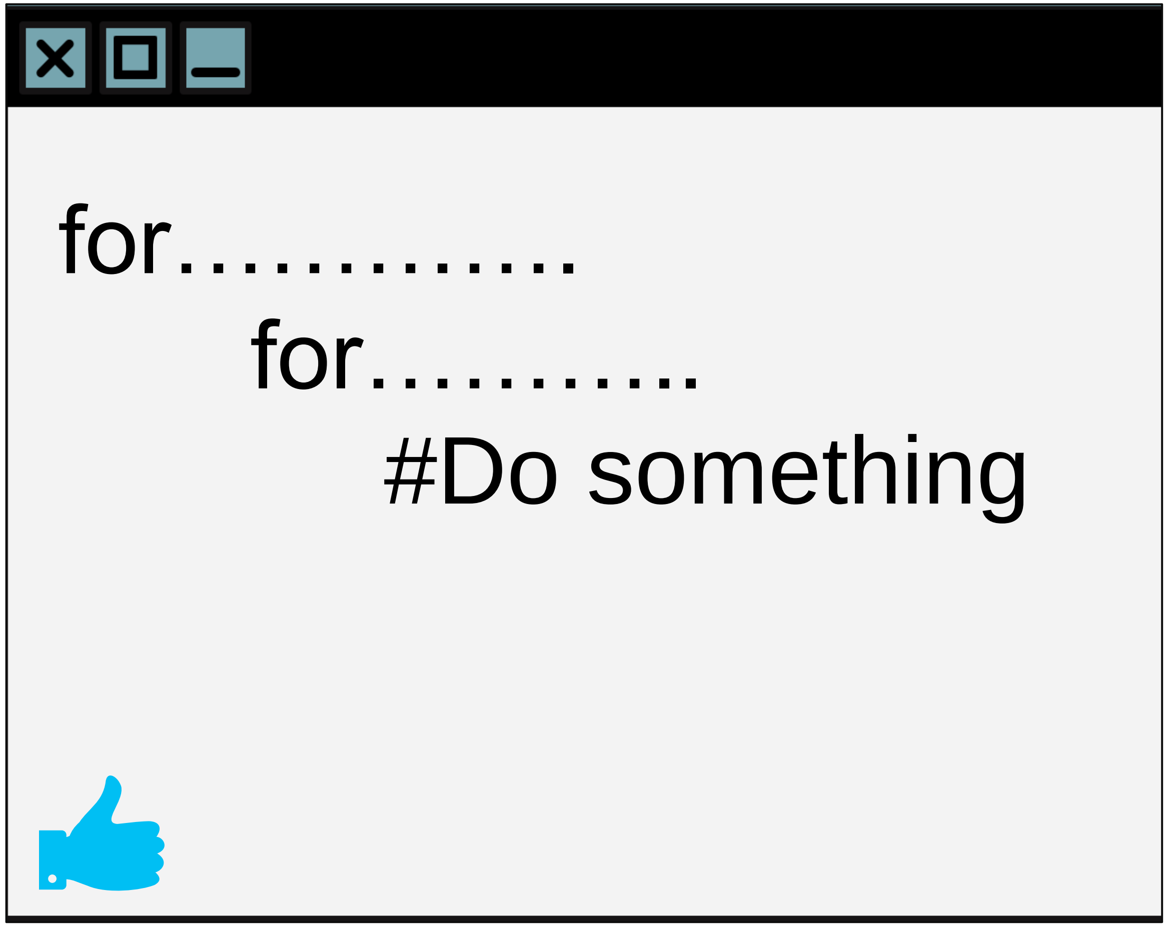
Problem 1

Going Home

$O(n^2)$

Algorithm

Using hash table

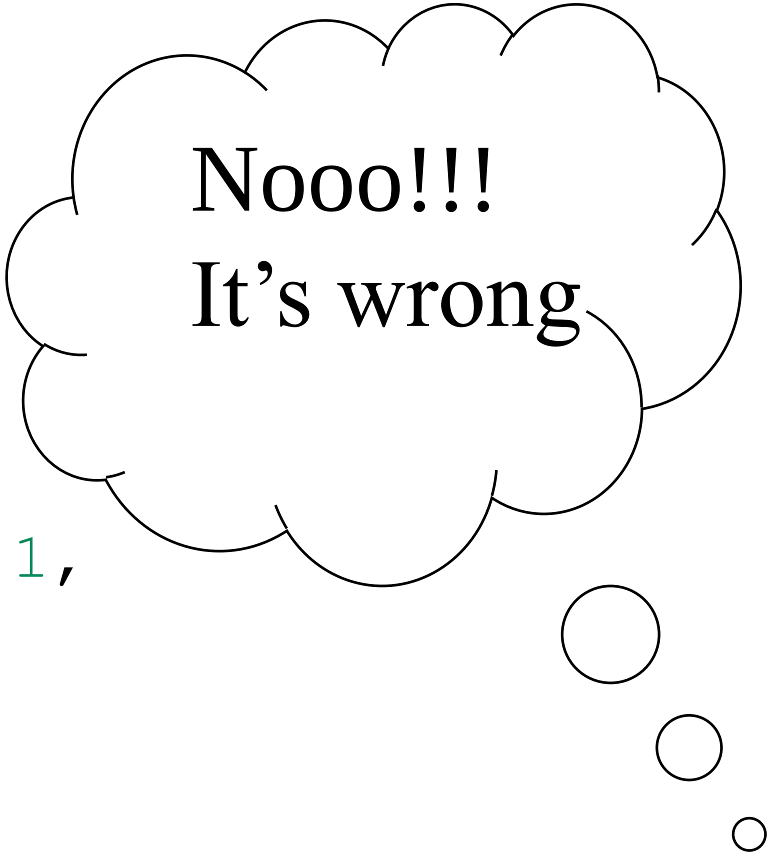


```
for.............
      for...........
            #Do something
```

Let’s write some code

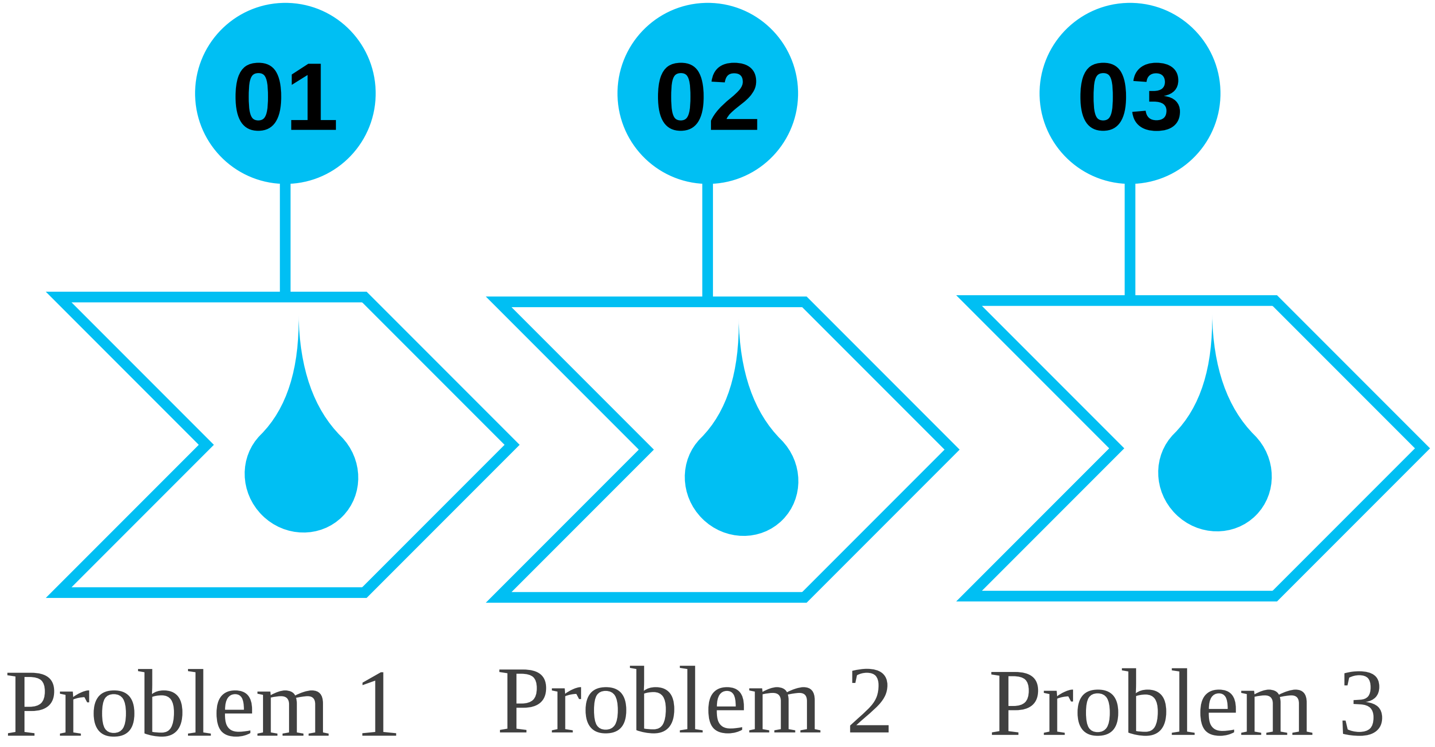
hashTable = { }

```
for first in range(size - 1):
   for second in range(first + 1, size):
       sum = numbers[first] + numbers[second]
       if sum in hashTable.keys():
         previous = hashTable.get(sum)
         print("YES")
         print(previous[0] + 1, previous[1]+ 1,
               first + 1, second + 1)
         return
       else:
            hashTable[sum] = (first, second)
    print("NO")
```



Nooo!!!
It’s wrong

Level's reward!!!



01

02

03

Problem 1

Problem 2

Problem 3

Problem 2

Allocate Books

Abstraction

Patter
Regcontion

Decomposition

Algorithm

Problem 2

Allocate Books

Abstraction

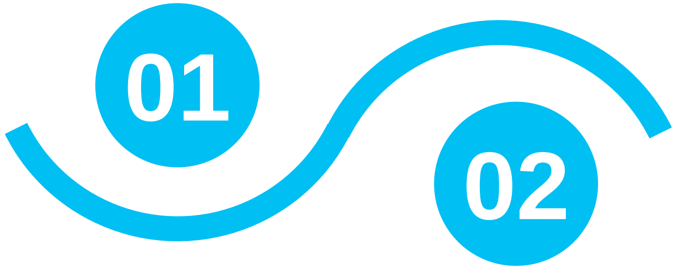
Assign elements in array in such a way that the maximum number of continuous-sum assigned to a group is minimum.

Problem 2

Allocate Books

Decomposition

Dive an array into different groups.



Find the minimun.

Problem 2

Allocate Books

Patter Regcontion

Array, in increasing order, find a value satisfied some condition???

Binary Search

Problem 2

Allocate Books

Algorithm

Binary Search

A function that check if a value is valid to apply BS or not.

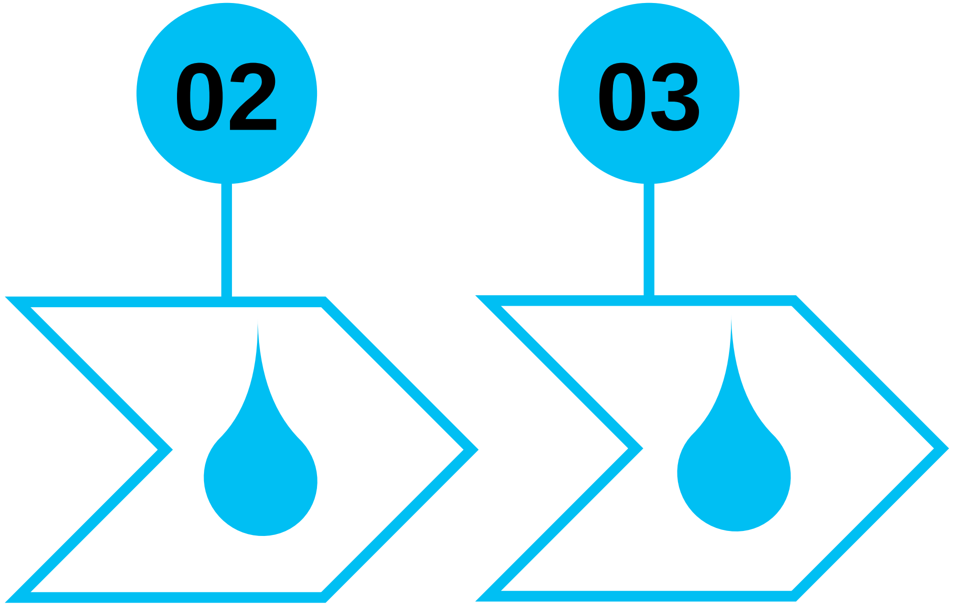
Let’s write some code

What should we do now?

```
def Check_If_Possible(value):
    allocatedStudents, pages = 1, 0
    for i in pagesList:
        if i > value:
            return False
        if pages + i > value:
            pages = i
            allocatedStudents += 1
        else:
            pages += i
    if allocatedStudents > numberOfStudents:
        return False
    return True
```

O(n*logn)

Level's reward!!!



02

03

Problem 2

Problem 3

Problem 3

Coins Change Problem

Abstraction

Patter Regcontion

Decomposition

Algorithm

Problem 3

Coins Change Problem

Abstraction

Find a group has number of elements is minimun whose sum equal a given number.

Problem 3

Coins Change Problem

Patter Regcontion

We can use one value many times, a larger value coin may equal to some smaller coins value…

It’s repeated!!!

Problem 3

Coins Change Problem

Algorithm

Dynamic Programming

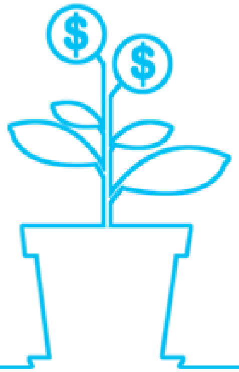
With an array can store result of already solved.

Let’s write some code

```
solution = [total + 1] * (total + 1)
solution[0] = 0
def FindSolutionForSubProblems():
    for index in range(1, total + 1):
        for coin in coinsList:
          if index - coin >= 0:
            solution[index] =
            min(solution[index], 1 + solution[index - coin])
    if solution[total] != total + 1:
        myAnswer = solution[total]
    else:
        myAnswer = -1
    return myAnswer
```

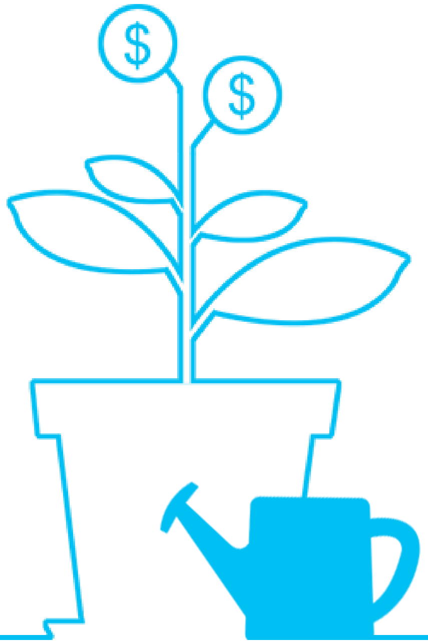
O(n*k)

Congratulation!!!
We made it!!!



Choose your reward

VS



Quiz

Thank you