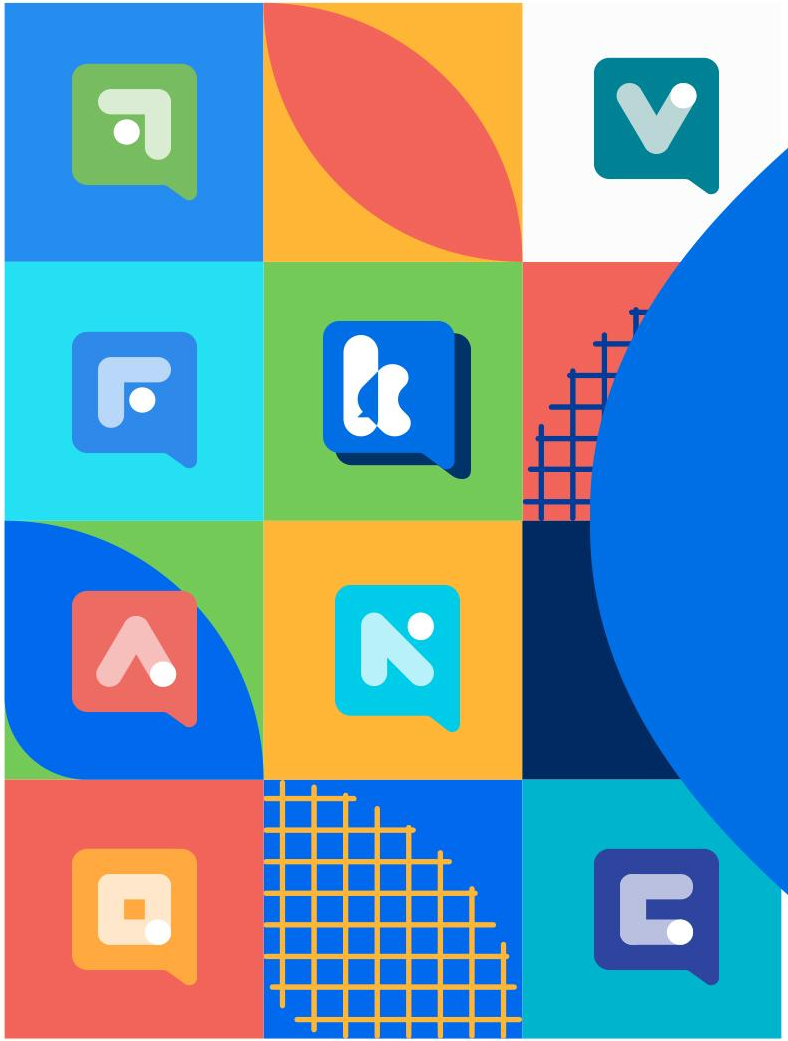


Rekursif

The important skill for CP

welly@kata.ai

Welly Tambunan

Head of Engineering and Data - kata.ai

Basic

Perkenalan terhadap rekursif

Prerequisite

- Looping
- Function

Motivasi

- Cool Guy

- Problem Solving Paradigm
  - Backtracking, Complete Search and Brute Force
  - DFS
  - Divide and Conquer
  - Dynamic Programming

Pengenalan

- Rekursi adalah sebuah fungsi menyelesaikan sebuah permasalahan dengan cara memanggil diri sendiri

- Jika masalah sudah cukup kecil, maka fungsi rekursi dapat langsung menghasilkan jawaban

- Jika masalah terlalu besar, maka fungsi akan memanggil diri sendiri dengan cakupan masalah yang lebih kecil.

Pentingnya Rekursi

- Banyak permasalahan yang lebih mudah diselesaikan dan pendek kodenya jika menggunakan pendekatan rekursif.
- Strategi iteratif (for loop) dan rekursif sama-sama melakukan sesuatu yang berulang-ulang.
- Namun, terkadang solusi iteratif untuk suatu masalah sangat sulit untuk dipikirkan dan memerlukan teknik khusus.
- Dengan solusi rekursif, mungkin saja lebih mudah untuk melihat dan merancang alur penyelesaiannya.

Strategi Rekursif

- Terdapat dua hal yang perlu dipikirkan ketika menggunakan strategi rekursif:
    - *Base case*
      Apa kasus paling sederhana dari permasalahan ini?
    - *Recurrence relation*
      Bagaimana hubungan rekursif dari persoalan ini dengan persoalan serupa yang lebih kecil?

Factorial

$$n! = n \times (n-1) \times (n-2) \times \ \ldots \ \ \times 2 \times 1$$

```
factorial(0) = 1

factorial(n) = n * factorial(n-1)    [for n>0]
```


TLX

CodingBat code practice

Java | Python

Recursion-1 ☆☆☆☆☆☆☆☆ chance

Basic recursion problems. Recursion strategy: first test for one or two base cases that are so simple, the answer can be returned immediately. Otherwise, make a recursive a call for a smaller case (that is, a case which is a step towards the base case). Assume that the recursive call works correctly, and fix up what it returns to make the answer.

- factorial H
- bunnyEars2
- count7
- countX
- changePi
- array11
- pairStar
- countAbc
- countHi2
- strCount
- bunnyEars H
- triangle
- count8
- countHi
- noX
- array220
- endX
- count11
- parenBit
- strCopies
- fibonacci
- sumDigits
- powerN
- changeXY
- array6
- allStar
- countPairs
- stringClean
- nestParen
- strDist

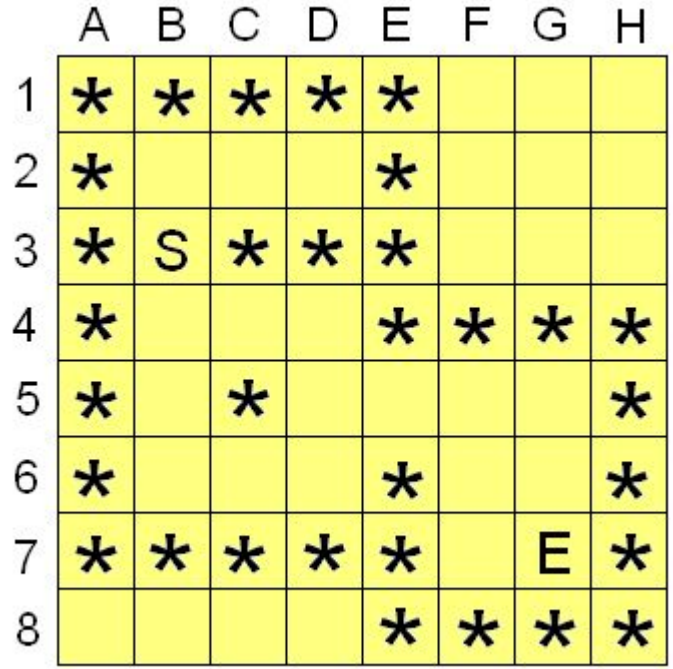
Kompleksitas

- Baik secara iteratif maupun rekursif, kompleksitasnya adalah O(N).
- Setiap pemanggilan rekursif membutuhkan *alokasi memori*, sehingga jika pemanggilannya semakin dalam, semakin banyak *tambahan memori* yang digunakan.
- Waktu untuk mengalokasikan memori juga menyebabkan solusi rekursif cenderung bekerja *lebih lambat* dibandingkan solusi iteratif.

Intermediate

Multiple related decision

Multiple Related Decisions



Must know !

- Permutasi
- Kombinasi
- Power Set
- Binary Search
- Merge Sort
- Quick Sort

Latihan

Advanced

Complete Search (Brute force and Backtracking)
DP (Top Down + Memo)

Fibonacci

```
Fibonacci[0] = 0
Fibonacci[1] = 1
Fibonacci[n] = Fibonacci[n - 2] + Fibonacci[n - 1]
```

Knapsack

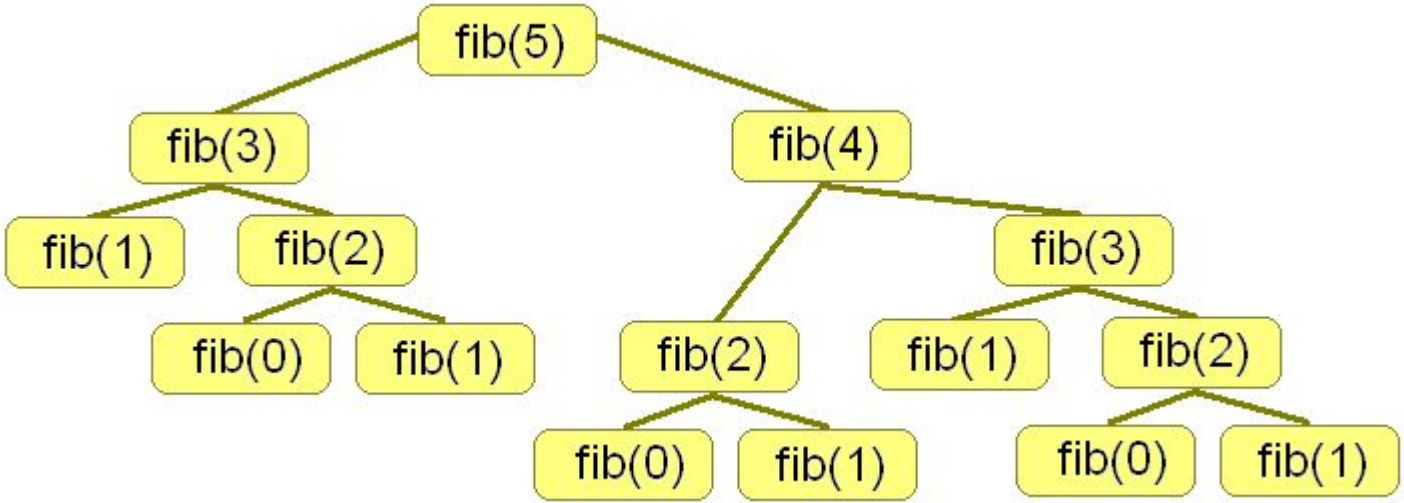
Given an array C[1..K] of distinct positive integers, count how many combinations of integers in C add up to exactly N.

Knapsack

Given an array of ints, is it possible to choose a group of some of the ints, such that the group sums to the given target?

groupSum(0, [2, 4, 8], 10) → true
groupSum(0, [2, 4, 8], 14) → true
groupSum(0, [2, 4, 8], 9) → false

Fibonacci


fib(5)
fib(3)
fib(4)
fib(1)
fib(2)
fib(3)
fib(0)
fib(1)
fib(2)
fib(1)
fib(2)
fib(0)
fib(1)
fib(0)
fib(1)

CodingBat code practice

Java | Python

Recursion-2 ☆☆☆ chance

Harder recursion problems. Currently, these are all recursive backtracking problems with arrays.

- groupSum H
- groupSum5
- splitOdd10
- groupSum6
- groupSumClump
- split53
- groupNoAdj
- splitArray