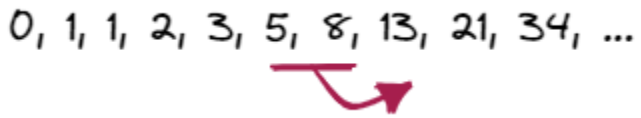
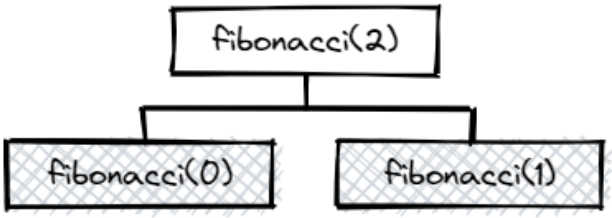
Time Complexity

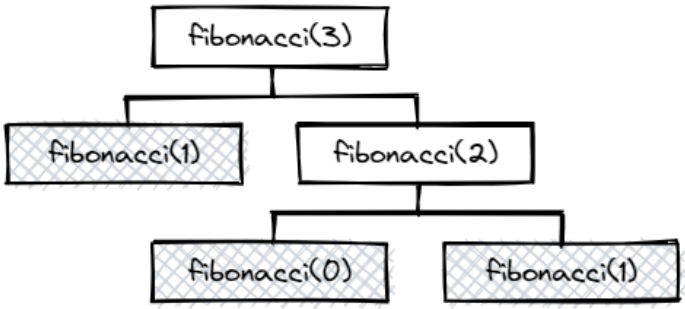
Let's consider a function that returns the nth fibonacci number



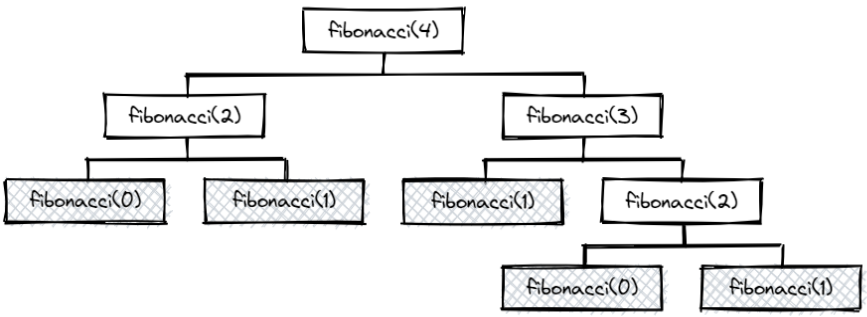

```
function fibonacci(n) {
  if (n <= 1) {
    return 1;
  }
  return fibonacci(n - 2) + fibonacci(n - 1);
}
```



NUMBER OF INSTRUCTIONS = 3

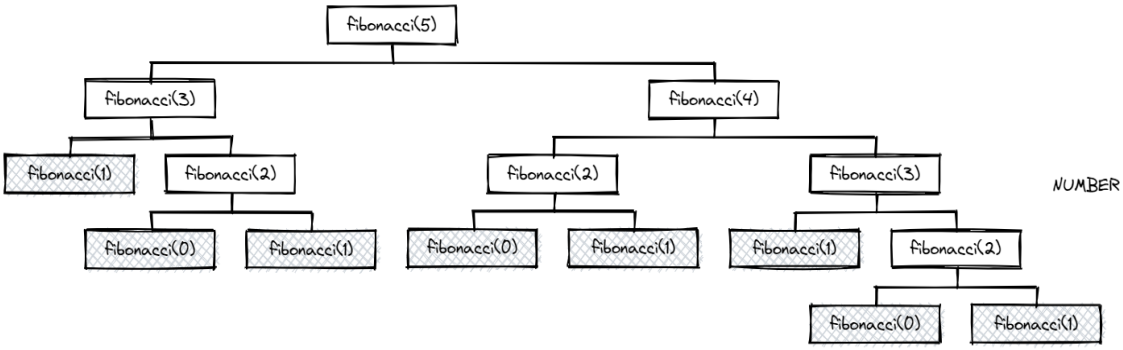


NUMBER OF INSTRUCTIONS = 5

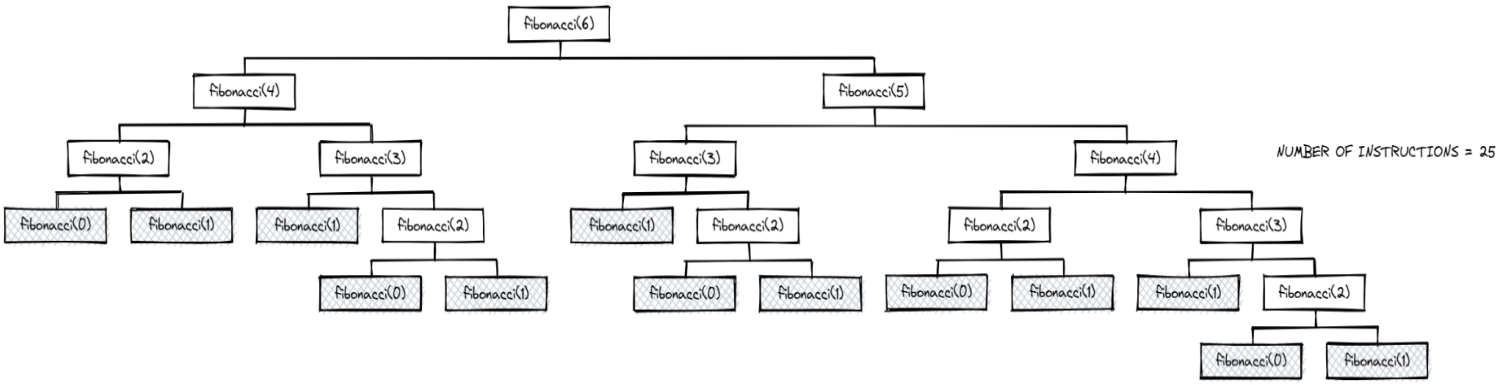


NUMBER OF INSTRUCTIONS = 9

Time Complexity

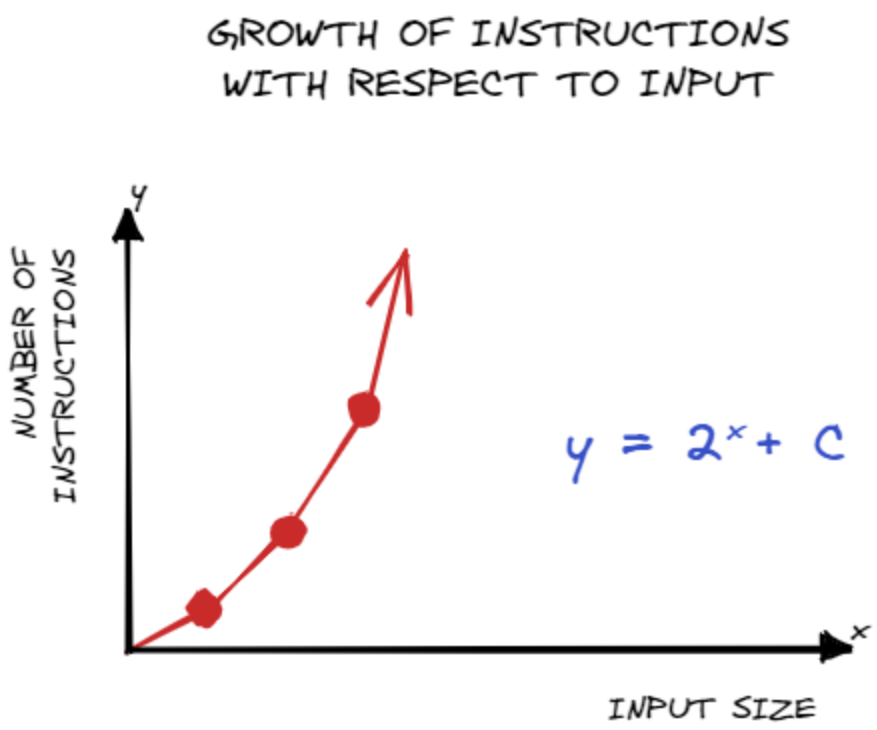


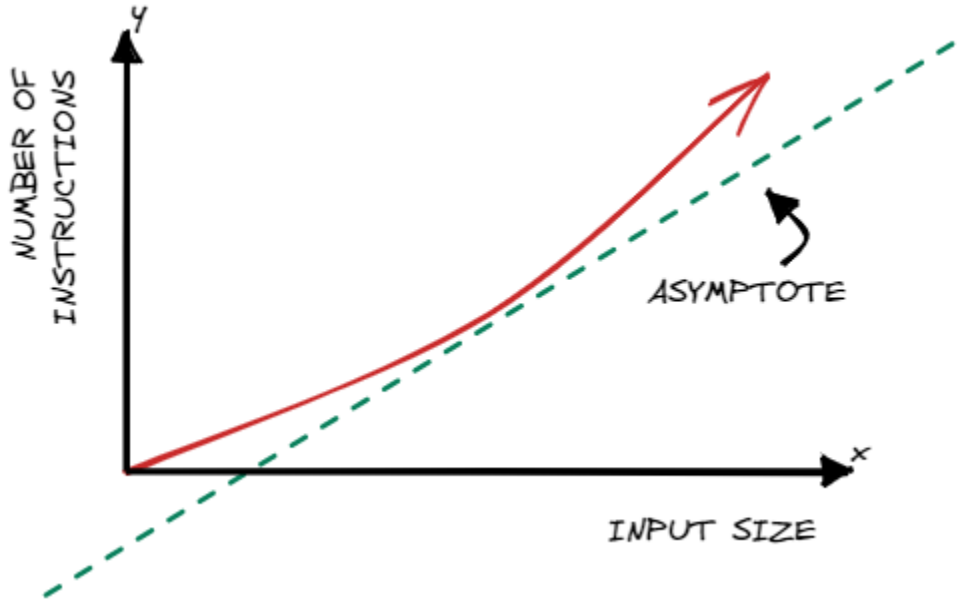
NUMBER OF INSTRUCTIONS = 15



NUMBER OF INSTRUCTIONS = 25

Time Complexity

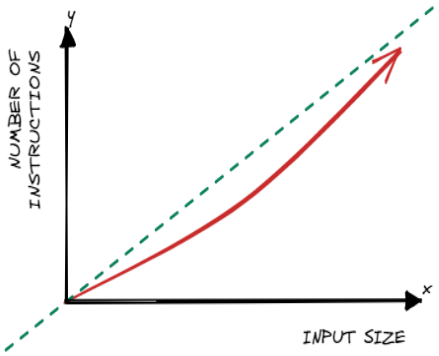




skill academy
by testbook

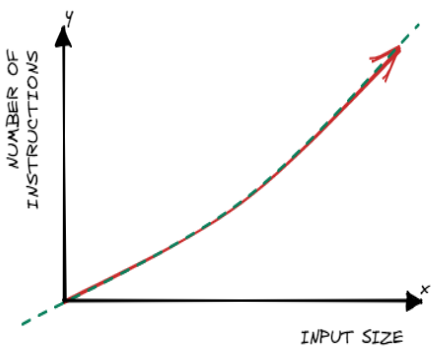
Time Complexity

O(n) -> ASYMPTOTIC UPPER BOUND

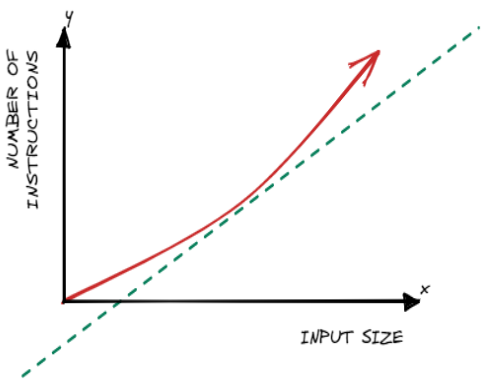


$O(2^n)$

θ(n) -> ASYMPTOTIC TIGHT BOUND



Ω(n) -> ASYMPTOTIC LOWER BOUND



$\Omega(n^2)$