**Fibonacci Numbers**

**Reading**

Computing Fibonacci numbers: Section 0.2 of [DPV08]

**If you find this lesson difficult to follow**

If you need to refresh your knowledge of recursion: [Section on recursion](https://www.khanacademy.org/computing/computer-science/algorithms/recursive-algorithms/a/recursion) at Algorithms class by Tom Cormen and Devin Balkcom at Khan Academy

**Visualizations**

[Computing Fibonacci numbers](http://www.cs.usfca.edu/~galles/visualization/DPFib.html) by David Galles

To better appreciate the difference between polynomial time and exponential time algorithms, try computing F20​ using this visualization. For this, enter "20" into the field and press "Fibonacci Recursive". This calls a recursive algorithm that makes an endless number of recursive calls. This call will never end even if you increase the visualization speed to maximum. Stop this call by pressing "Skip Forward" and press "Fibonacci Table". This will call an iterative algorithm that uses an array to compute Fibonacci numbers efficiently. The third button calls a recursive algorithm with memoization. We will cover such algorithms in the Dynamic Programming module later in this class.

(Note that the visualization uses a slightly different definition of Fibonacci numbers: there, F0=F1=1, and in the lecture, F0=0, F1=1. This, of course, has no influence on the running time.)

**Advanced Reading**

Properties of Fibonacci numbers: Exercises 0.2–0.4 in [DPV08]

**References**

[DPV] Sanjoy Dasgupta, Christos Papadimitriou, and Umesh Vazirani. Algorithms (1st Edition). McGraw-Hill Higher Education. 2008.

**GCD**

## Reading

Greatest common divisor: Section 1.2.3 of [DPV08], Section 31.2 of [CLRS]

## If you find this lesson difficult to follow

An [elementary introduction to greatest common divisor](https://www.khanacademy.org/math/pre-algebra/factors-multiples/greatest_common_divisor/v/greatest-common-divisor-factor-exercise) at Khan Academy

## References

[DPV] Sanjoy Dasgupta, Christos Papadimitriou, and Umesh Vazirani. Algorithms (1st Edition). McGraw-Hill Higher Education. 2008.

[CLRS] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms (3rd Edition). MIT Press and McGraw-Hill. 2009.

**Big-*O* Notation**

## Reading

Big-O notation and growth rate: Section 0.3 of [DPV08]

[Big-O notation](https://www.khanacademy.org/computing/computer-science/algorithms/asymptotic-notation/a/asymptotic-notation) at Khan Academy

## If you find this lesson difficult to follow

If you need to refresh your knowledge of logarithms: an [elementary introduction to logarithms](https://www.khanacademy.org/math/algebra2/exponential-and-logarithmic-functions/introduction-to-logarithms/a/intro-to-logarithms) at Khan Academy

## References

[DPV] Sanjoy Dasgupta, Christos Papadimitriou, and Umesh Vazirani. Algorithms (1st Edition). McGraw-Hill Higher Education. 2008.