

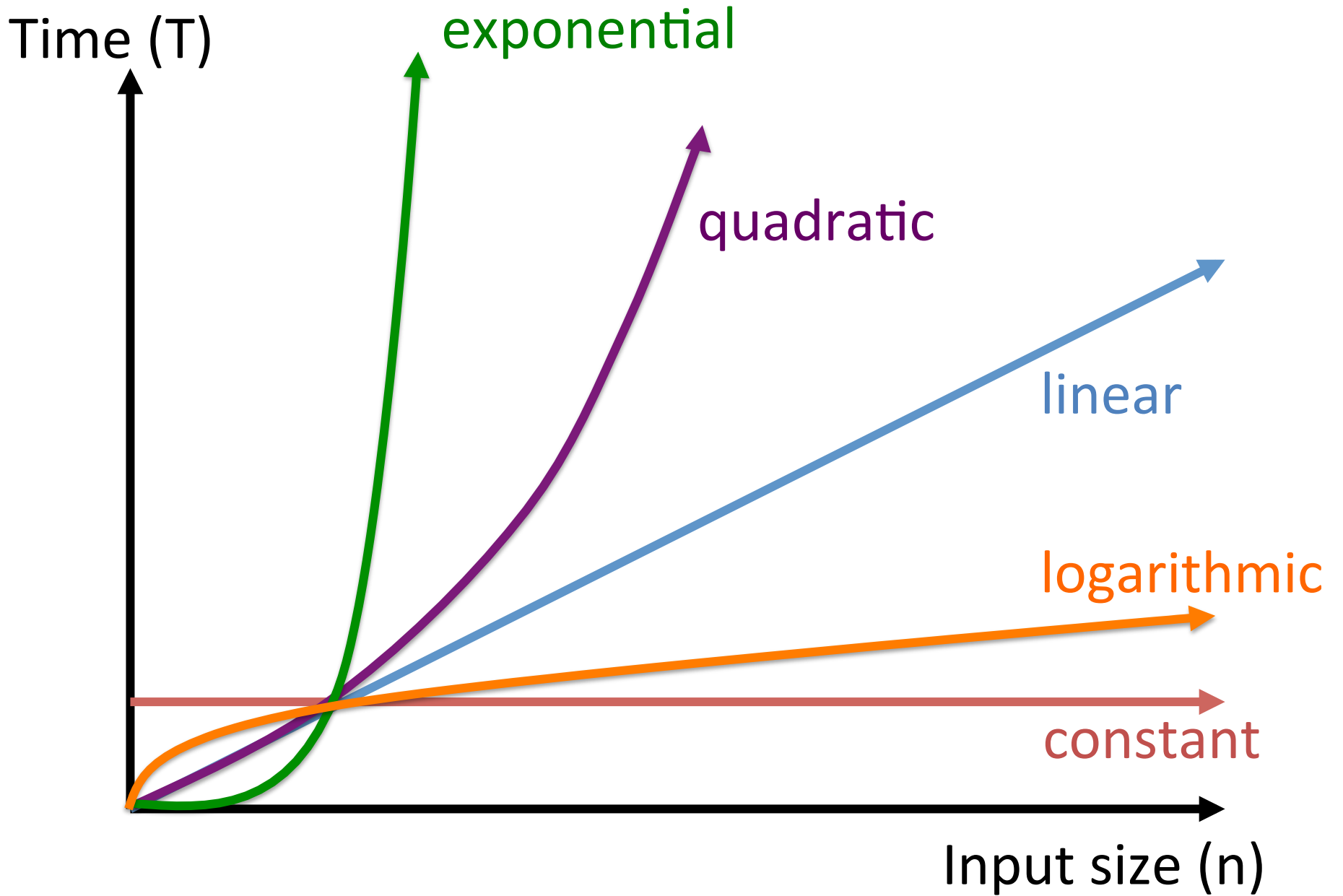
Linked Lists

Running time of algorithms

- We measure how fast an algorithm runs by calculating how much it slows down when we change the ***input size*** (amount of data the algorithm has to process).
- Usually we care about the ***worst possible running time*** (we're pessimistic!)

- If an algorithm's running time does not change when the size of the algorithm's input changes, the algorithm runs in ***constant time***.
- If the running time grows proportionally to the size of the input size, the algorithm runs in **linear time**.
- If the running time grows proportionally to the square of the input size, the algorithm runs in **quadratic time**.

- If the running time grows proportionally to some number raised to the power of the input size, the algorithm runs in **exponential time**.
 - Finding the n'th term of the Fibonacci sequence (using the double recursion formula) is exponential.
- What is binary search's running time?



One million “basic” operations per second.

	logarithmic	linear	quadratic	exponential
n = 10	0.0033 ms	0.01 ms	0.1 ms	1.024 ms
n = 20	0.0043 ms	0.02 ms	0.4 ms	1.049 sec
n = 30	0.0049 ms	0.03 ms	0.9 ms	17.9 min
n = 50	0.0056 ms	0.05 ms	2.5 ms	35.7 years
n = 100	0.0066 ms	0.1 ms	0.01 sec	4×10^{16} years
n = 1,000	0.0099 ms	1 ms	1 sec	3×10^{287} years
n = 10,000	0.0133 ms	10 ms	1.67 min	----
n = 100,000	0.0166 ms	0.1 sec	2.77 hours	----
n = 1,000,000	0.0199 ms	1 sec	11.57 days	----