Jamie Fravel
jfravel@iuuk.mff.cuni.cz

Asymptotics

1. Comparing functions Determine the asymptotic relationships between each pair of functions f and g in the following table (i.e. whether $f \in \mathcal{O}(g)$, $f \in \Omega(g)$ or $f = \Theta(g)$).

	$f(n)$	$g(n)$	$\mathcal{O}$	Ω	Θ
(1)	$n^2 - 30n + 5$	$0.7n^2 - 20n + 15$	○	○	○
(2)	$100n^3 + 40n^2 - n$	$0.5n^4 - 1000n^3$	○	○	○
(3)	$5n^2 - n$	$30n + 4$	○	○	○
(4)	n	$\sqrt{n}$	○	○	○
(5)	$n^{\frac{3}{4}}$	$\sqrt{n}$	○	○	○
(6)	$\log_2 n$	$\ln n$	○	○	○
(7)	$n(\log_2 n)^5$	$n\sqrt{n}$	○	○	○
(8)	2^n	2^{2n}	○	○	○
(9)	e^n	2^n	○	○	○
(10)	$n!$	n^n	○	○	○
(11)	$n!$	2^n	○	○	○
(12)	2^n	2^{n+1}	○	○	○

Lets start with a few different definitions of $\mathcal{O}$, Ω and Θ.

Intuitively, $f \in \mathcal{O}(g)$ if f grows at *most* as fast as g; $f \in \Omega(g)$ if f grows at most as *least* as g and $f \in \Theta(g)$ if f grows *about* as fast as g.

Another intuition could be $f \in \mathcal{O}(g)$ if $g(n)$ never becomes irrelevantly small when compared to $f(n)$; $f \in \Omega(g)$ if $f(n)$ never becomes irrelevantly small when compared to $g(n)$; grows at *most* as fast as g and $f \in \theta(g)$ if f and g never become irrelevant compared to each other.

Formally, $\mathcal{O}$, Ω and Θ are defined:

- $f \in \mathcal{O}(g)$ if there exist constants $c, m > 0$ such that $f(n) \leq cg(n)$ for all $n > m$.
- $f \in \Omega(g)$ if there exist constants $c, m > 0$ such that $f(n) \geq cg(n)$ for all $n > m$.
- $f \in \Theta(g)$ if $f \in \mathcal{O}(g)$ and $f \in \Omega(g)$. That is $\Theta = \mathcal{O} \cap \Omega$.

These are the definitions we expect you to know and use for this class. However, it is often useful to consider a slight alteration of these definitions:

- $f \in \mathcal{O}(g)$ if $\frac{f(n)}{g(n)}$ is bounded from above as n increases (as $n \to \infty$).
- $f \in \Omega(g)$ if $\frac{f(n)}{g(n)}$ remains strictly positive as n increases (as $n \to \infty$).
- $f \in \Theta(g)$ if $\frac{f(n)}{g(n)} \in (0, \infty)$ as n increases (as $n \to \infty$).

These are equivalent when you notice that c should be a real, finite, positive number.

2. Properties of asymptotic notation Let $f(n)$ and $g(n)$ be asymptotically positive functions (which means that $\lim_{n\to\infty} f(n) > 0$ and $\lim_{n\to\infty} g(n) > 0$). Prove or disprove each of the following conjectures

(1) $f(n) + g(n) \in \Theta(\min(f(n), g(n)))$.

(2) $f(n) + g(n) \in \Theta(\max(f(n), g(n)))$.

(3) $f(n) \in \mathcal{O}((f(n))^2)$.

(4) $f(n) \in \Omega((f(n))^2)$.

(5) $f(n) \in \mathcal{O}(g(n))$ implies $g(n) \in \Omega(f(n))$.

2nd tutorial (continued)

If the tree (and hence tree edges) represent a single path from the root to every other vertex, which edges constitute a second path?

Interlude: Bridges. Jamie will use SYGA to demo an algorithm for finding "bridges" in undirected graphs. See page 122 of https://iuuk.mff.cuni.cz/ koutecky/pruvodce-en-wip.pdf for a reference. Note that translation of this document into English is still a work in progress.
This also serves as our introduction to the SYGA (See Your Graph Algorithm) visualizer.

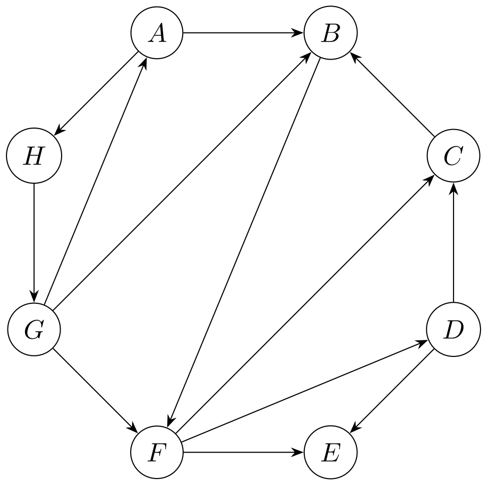
Depth First Search

3. DFS Runs — SYGA Starting at vertex A, perform depth-first search on the following graph in SYGA:

syga.space/exercises/week-2/practice-depth-first-search.

Whenever there's a choice of vertices, proceed in alphabetical order. First, find the tree edges, then classify the non-tree edges into back, forward and cross edges.

Finally you can compare your solution to one generated algorithmically. Feel free to change the graph (at the top of the script) or explore the algorithm.



4. Singly Connected Graphs — SYGA A directed graph G is *singly connected* if for every pair of vertices u and v there is at most one path from u to v in G. Follow this link to the SYGA exercise:

syga.space/exercise/week-2/singly-connected-graphs

Adjust the algorithm to determine whether or not a directed graph is singly connected. Your finished algorithm should set `flag = 1` whenever you find a counterexample.

There are three graphs at the top of the script for you to try your solution on. The first two are *not* singly connected.

> Tree edges represent one path from the root to every other (reachable) node in the graph. What kinds of non-tree edges are there?

5. Longest Path In a DAG Given a directed, acyclic graph (DAG) $G = (V, E)$ and two of its vertices s and t, we want to compute the length of a longest path from s to t.

> **Proposition 1.1.** *Running DFS on an acyclic graph will never produce a back edge.*
>
> Back edges necessarily imply a cycle.

6. Count Paths In a DAG Given a DAG $G = (V, E)$ and two of its vertices s and t, design an algorithm that calculates the number of distinct paths from s to t.