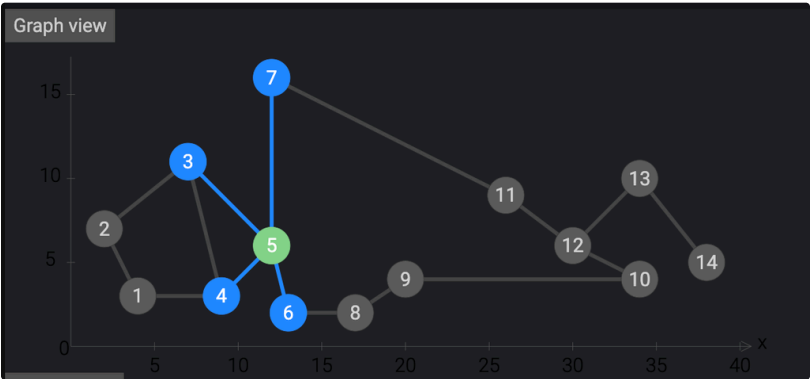# Colour guidelines

Colours will be given a specific meaning for each type of visualisation. When adding a new visualisation make sure to adhere to the preexisting conventions for the meaning of each colour.

Graph:

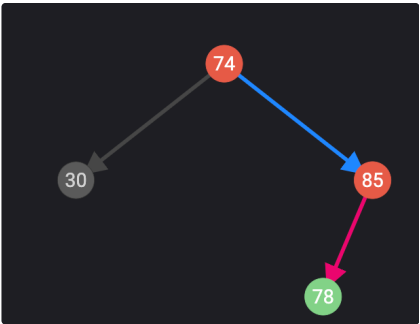Leaf (green) will represent the selected node

Sky (blue) will represent available/adjacent nodes

Apple (red) will represent a selected edge

Peach (orange) is currently unused

Typically graphs use the Array2DTracer.js and Array2DRenderer.module.scss files.

Tree:

Leaf - leaf nodes (fitting)

Sky - edge from parent node to parent node

Apple - edge from parent node to leaf node

Peach - parent node

Uses GraphTracer.js and GraphRenderer.module.scss

Table:

Leaf - "Found" table elements in a search function, or sorted elements
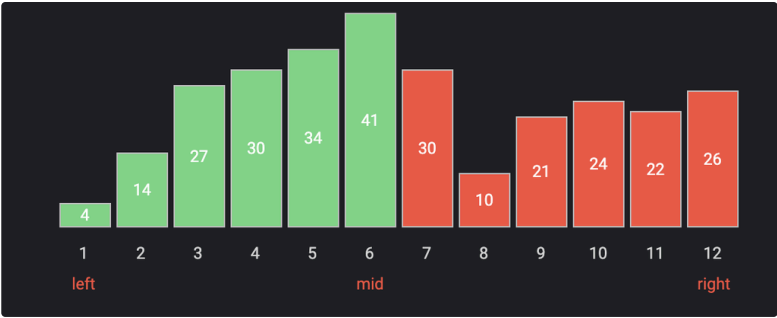
Sky - selected elements

Apple - unused

Peach - unused

Also uses Array2DTracer.js and Array2DRenderer.module.scss

Array:

Leaf - sorted elements

Sky - unused

Apple - unused

Peach - unsorted elements

Uses Array1DTraces.js and Array1DRenderer.module.scss

Other:

a	b	c	d	e	f	g	h	i	j	k	l	m
1	4	4	2	4	4	4	4	4	4	4	4	4

n	o	p	q	r	s	t	u	v	w	x	y	z	space
4	4	4	4	4	4	4	4	4	4	4	4	4	4

Horspool String Search

d	d	a	c

d	c	a	c	c	d	d	d	a	b	d	d	a	c

Horspool's algorithm uses TwoArray2DTracer.js and TwoArray2DRenderer.module.scss, and seems to be the only algorithm that does so. If you choose to create new visualisations in the same style, update the style guidelines to ensure your visualisation is consistent with Horspool's algorithm

To add a new colour to a certain type of visualisation:

This will need to be done by adding a new scss declaration in the renderer module for the type of visualisation, and updating the associated tracer and index.js files found in src/components/datastructures to include those colours
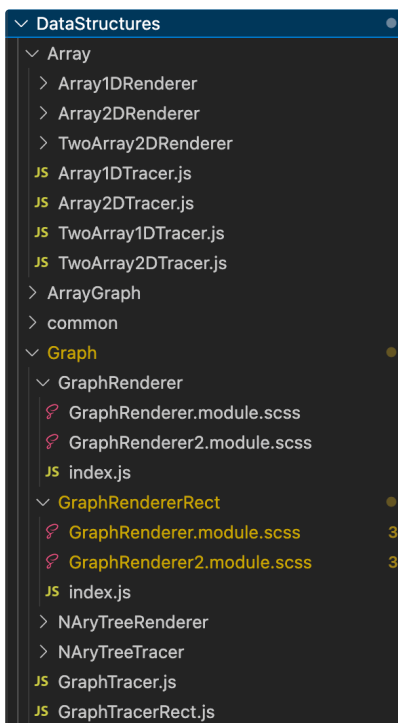
If at all avoidable, do not add a new colour to the 4 main colours in each colour scheme, in order to keep things simple and consistent. Most visualisation types still do not use all four main colours, so these should be used before any others