



# Graph Theory 2

...

Shortest Path

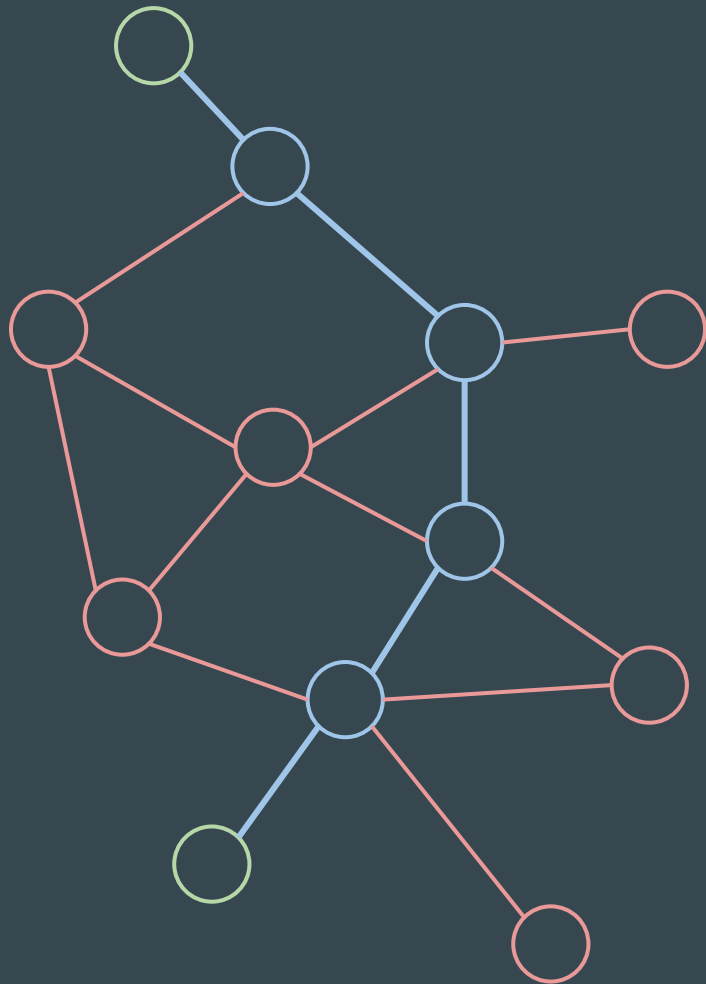
# Simple case

We are given a connected graph which edges all have the same weight

→ find the shortest path that connects two given nodes

(on this graph, in blue - its length is 5)

/!\ Multiple shortest paths are possible



# Breadth-first search

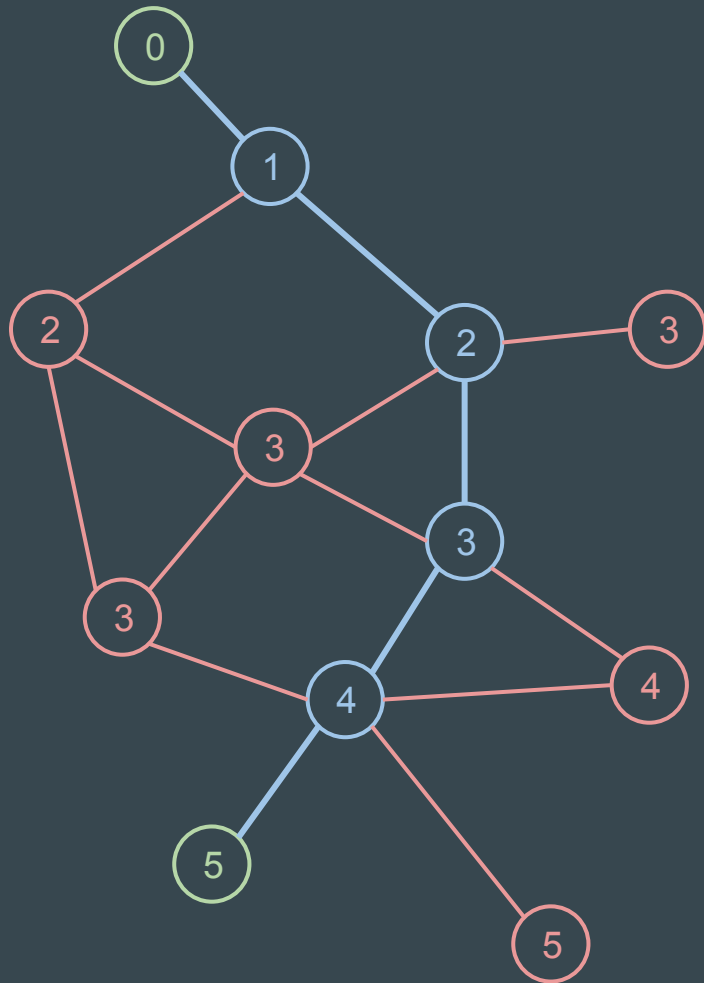
Put one end of the path in a queue and iterate:

- dequeue one element
- enqueue all unvisited neighbors and mark the element as their parent

Then, from the other end of the path, follow the parent-child relationships

(the nodes are visited in order of distance from the starting node, as shown on the right)

→ Complexity :  $O(|E| + |V|)$



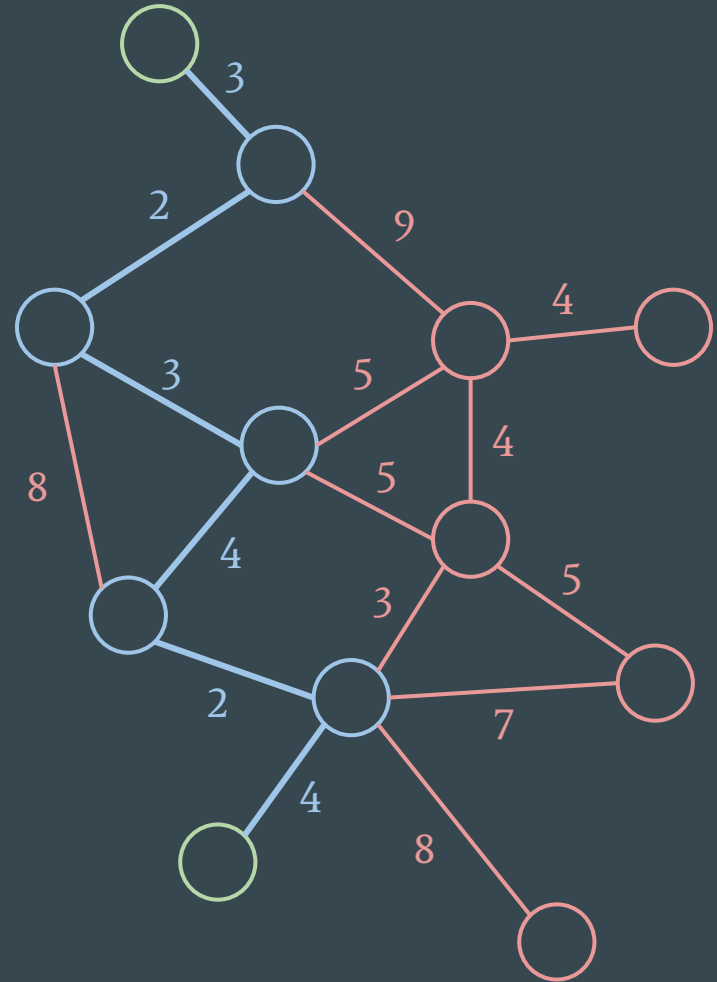
# Harder case: weighted edges

Now the edges have positive weights

→ we want to find the path which edges have the smallest sum, between two given nodes

(on this graph, this sum is 18)

/!\ Multiple best paths are possible

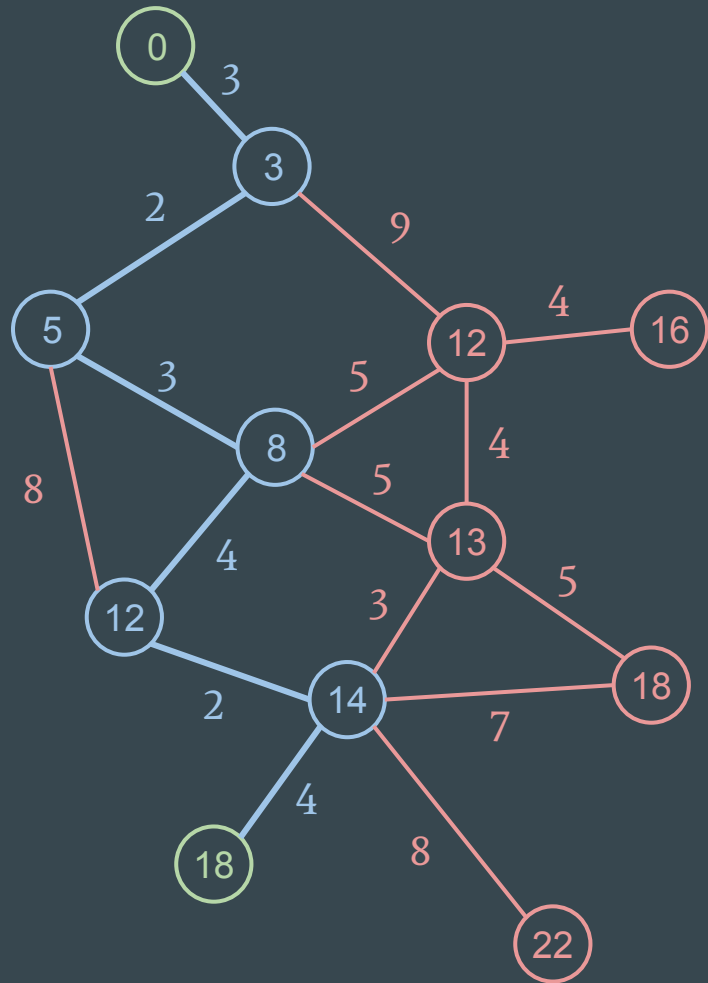


# Dijkstra's algorithm

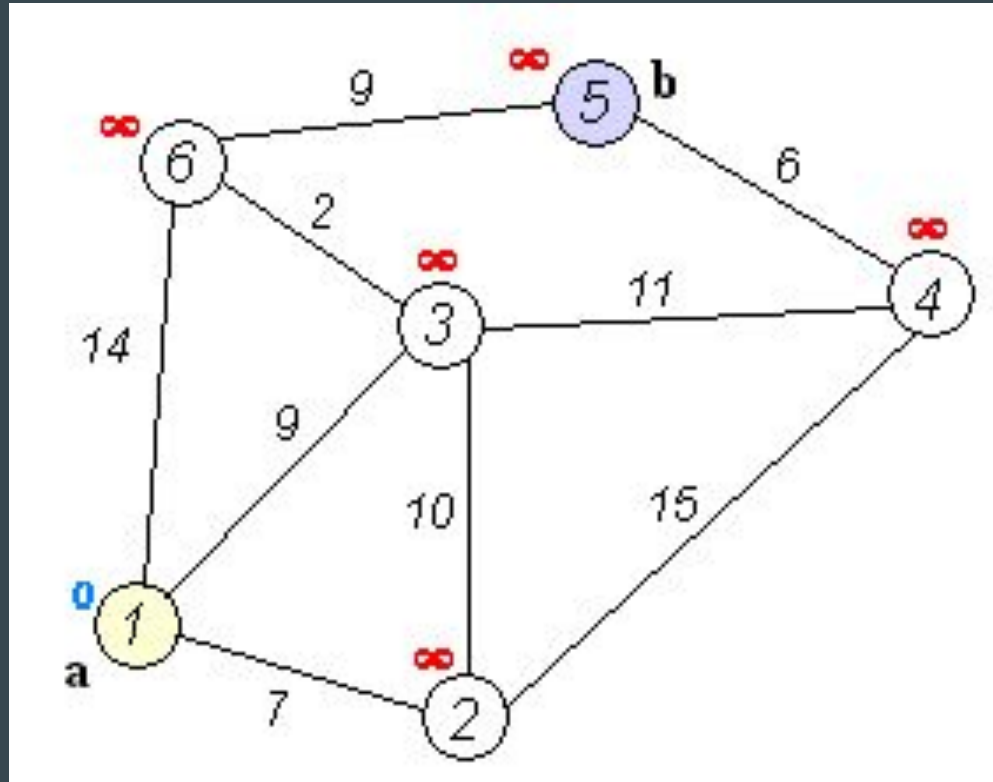
- Works only if the edges have positive weights
- It's a same idea than BFS
- This time the nodes are added to a heap queue so that you always check the closest to the starting node

(on the right, the distance to the starting node is written on every node)

→ Complexity :  $O(|E| + |V| \log |V|)$



# Dijkstra's algorithm : stolen gif from Wikipedia



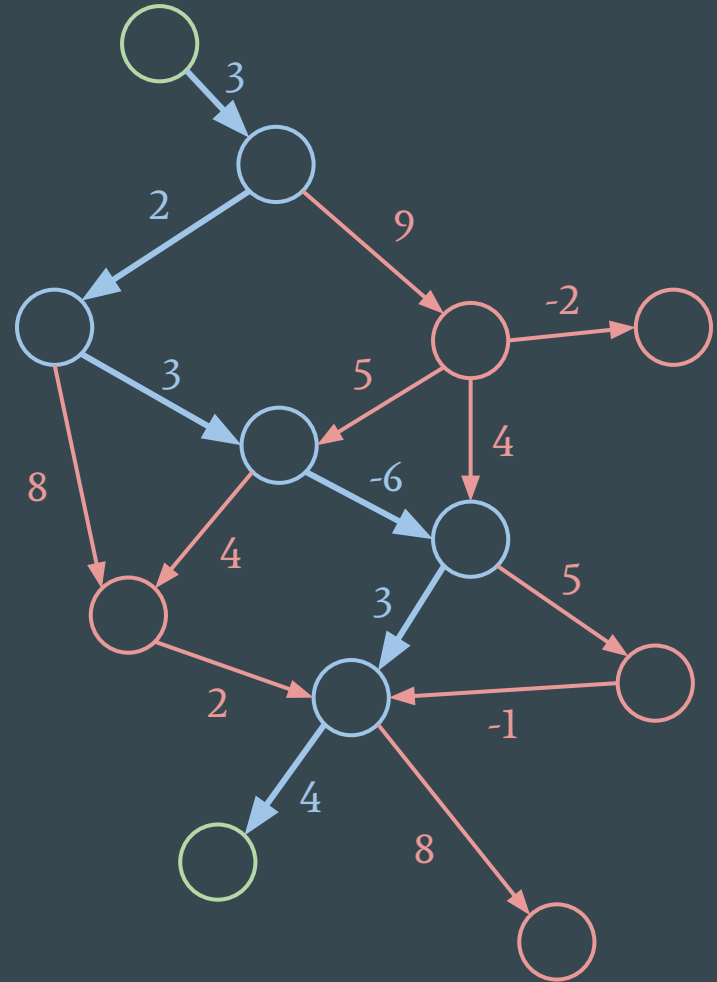
# Harder case: with negative edges

Now the edges can have positive or negative weights

This only works on directed graphs, because a negative edge on an undirected graph means a negative cycle and there is no solution

(on this graph, this sum is 9)

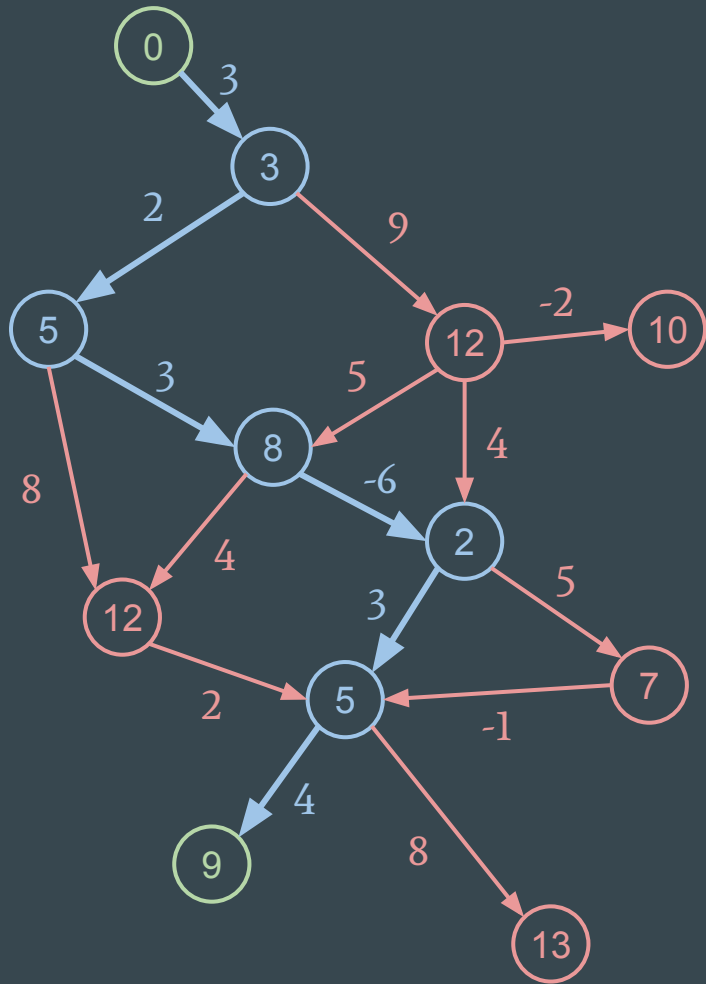
/!\ Multiple best paths are possible



# Bellman-Ford algorithm

- This algorithm marks all the nodes as infinitely far from the starting node
- It then executes  $|V|$  *relaxations*
- A *relaxation* browses each edge to check if it can improve the current distance of one of its ends to the center
- If at the end, progress can still be made, the graph contains a negative cycle

→ on GitHub, see [bellman-ford.py](#) ;  $O(|E| * |V|)$





# More algorithms

- Depth-first search (in a tree):
- Floyd-Warshall (shortest path between any pair of nodes): [https://en.wikipedia.org/wiki/Floyd\\_Warshall](https://en.wikipedia.org/wiki/Floyd_Warshall)
- A\* (extension of Dijkstra with heuristics):  
[https://en.wikipedia.org/wiki/A\\*\\_search\\_algorithm](https://en.wikipedia.org/wiki/A*_search_algorithm) -  
we will probably talk about it later

# Goodbye

It's the last course I'm teaching because  
I'm moving to **Scotland** on wednesday

Good bye, and the adventure with  
INSAIgo continues! ;)



# Credits

Slides: Louis Sugy for INSAIgo

Pictures of Scotland from Pexels - <https://www.pexels.com>