

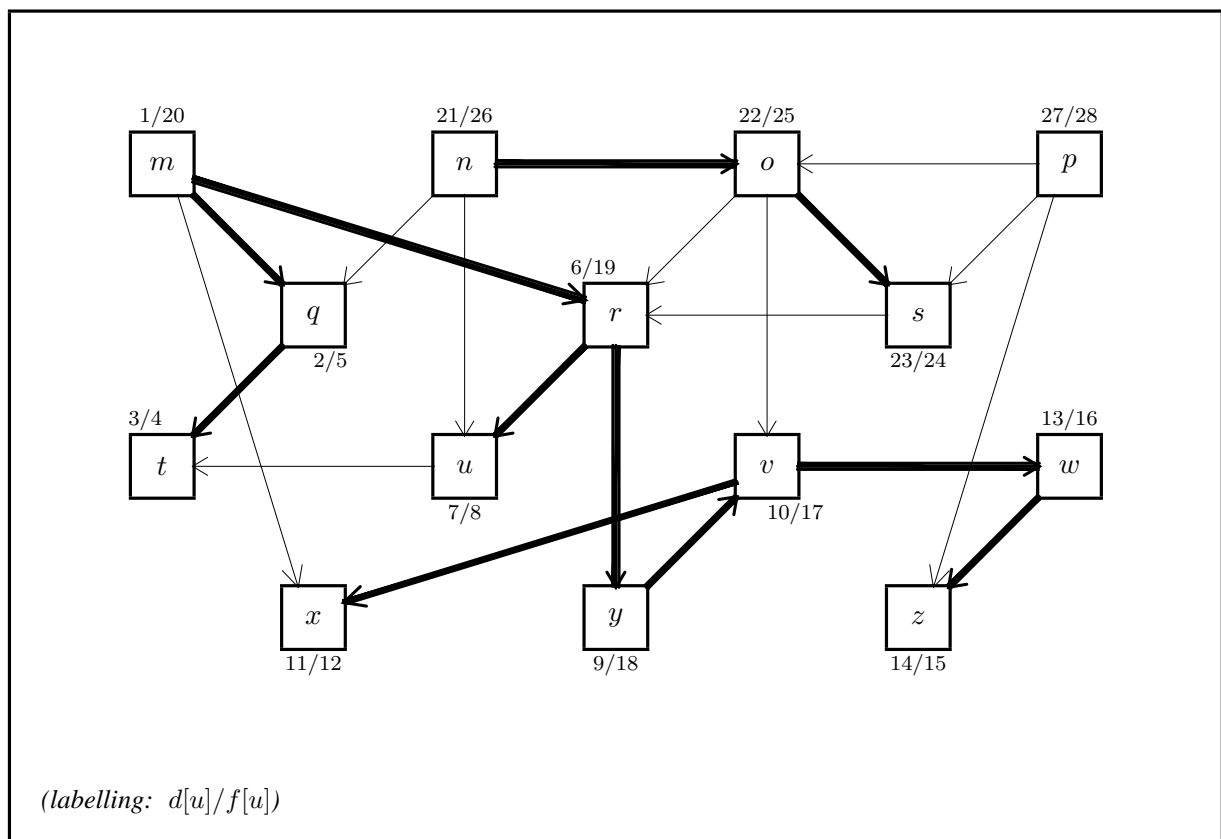
1 General remarks

This week is about

- digraphs
- BFS and DFS.

2 An old acquaintance

Consider the digraph G we have already seen in the lectures:



- Q1** Relabel the vertices, so that they then carry the numbers $1, \dots, 14$, in ascending order of discovery times, obtaining G' .
- Q2** Just rerun DFS on G' , following numerical order, checking that the given discovery and finishing times are correct. Make sure that you understand the jumps when backtracking. Which path corresponds to the largest size of the stack (that is, the longest path from the root to some vertex)?
- Q3** Read off the topological sorting of G' .
- Q4** Check that we have actually a topological sorting, by considering each vertex v and its out-neighbours w_1, \dots, w_m , and making sure that indeed all w_i are to the right of v in the sorting.
How many checks are needed?

- Q5** Consider the digraph G'' obtained from G' by adding one arc from vertex 10 to vertex 1. What changes, when you run DFS on G'' ? Extract a topological sorting from that run of DFS on G'' and check it — what happens?
- Q6** Run DFS on G'' in reverse numerical order.
- Q7** Obtain a possible topological order from it. Where is now the first fault, when you run through the vertices in numerical order?
- Q8** Finally run BFS on G'' with start vertex 14, in numerical order (as usual for BFS, without restart). Show the obtained directed BFS-tree with root 14. Which vertices are not reachable from vertex 14? What is the longest distance from vertex 14 to any reachable node? List all the alternative shortest routes (paths) to the vertices farthest away.