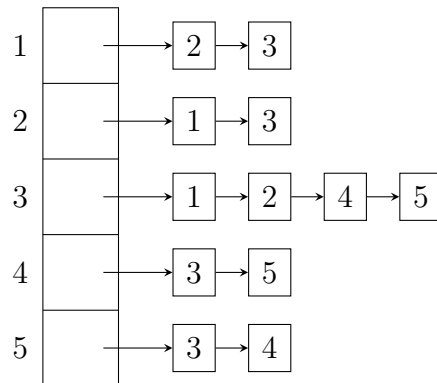
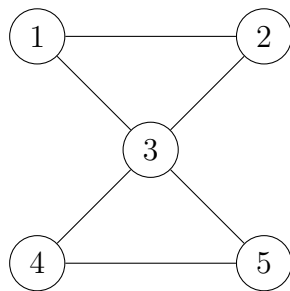


Fundamental Algorithms

Exercise Sheet 8

1. A graph can be represented by *adjacency lists*, where we keep a list of all adjacent vertices for each node. For instance, the following graph on the left can be represented by the lists on the right. The leftmost structure of the lists is simply an array of pointers to the lists.



- Given a graph $G = (V, E)$ in adjacency list representation, write down an algorithm with the running time $\mathcal{O}(|E|)$ that finds an Euler circle in the graph if one exists. Design first a data structure for adjacency lists.
2. What is wrong with the following argument? Let G be a graph with n nodes. When a node v of G is visited during a depth-first search, we immediately explore all other nodes that can be reached from v by following edges in G . In the topological sorting, these other nodes must come earlier than v . Thus, to obtain a topological sorting of the nodes, it suffices to use the algorithm for pre-order numbering (slide page 120), and reverse the order of the result (by subtracting the numbering of each node from $n + 1$).
3. In the *knapsack* problem, we are given a knapsack, a weight W , and n types of objects, numbered 1 through n . Assume that there are enough objects of each type available. Each object type $i \in \{1, \dots, n\}$ has a positive weight w_i and a positive value v_i . The knapsack can only be filled with objects whose total weight does not exceed W . The aim is to fill the knapsack in a way that maximizes the value of the included objects. It is not allowed to take a fraction of an object.

Design an algorithm that solves the knapsack problem by outputting the maximum value. An easy solution is to view the problem as a search in a directed graph, where nodes contain information about objects in the knapsack. An edge in the graph corresponds to an action of putting an object into the knapsack. The search is started from the node that represents the empty knapsack, and repeats putting new objects into it. Notice that there is no need to construct the graph explicitly.