State-Space Graphs
• There are various methods for searching state spaces.
• One possibility is breadth-first search:
  – Mark the start node with a 0.
  – Mark all adjacent nodes with 1.
  – Mark all unmarked nodes adjacent to nodes with a 1 with the number 2, and so on, until you arrive at a goal node.
  – Finally, trace a path back from the goal to the start along a sequence of decreasing numbers.

Decision Trees
A decision tree is a special case of a state-space graph.
It is a rooted tree in which each internal node corresponds to a decision, with a subtree at these nodes for each possible outcome of the decision.
Decision trees can be used to model problems in which a series of decisions leads to a solution.
The possible solutions of the problem correspond to the paths from the root to the leaves of the decision tree.

Example: The n-queens problem
How can we place n queens on an n×n chessboard so that no two queens can capture each other?
A queen can move any number of squares horizontally, vertically, and diagonally.
Here, the possible target squares of the queen Q are marked with an x.

Obviously, in any solution of the n-queens problem, there must be exactly one queen in each column of the board.
Otherwise, the two queens in the same column could capture each other.
Therefore, we can describe the solution of this problem as a sequence of n decisions:
Decision 1: Place a queen in the first column.
Decision 2: Place a queen in the second column.
... Decision n: Place a queen in the n-th column.
This way, "only" 256 configurations need to be tested.

Backtracking in Decision Trees
• There are problems that require us to perform an exhaustive search of all possible sequences of decisions in order to find the solution.
• We can solve such problems by constructing the complete decision tree and then find a path from its root to a leave that corresponds to a solution of the problem (breadth-first search often requires the construction of an almost complete decision tree).
• In many cases, the efficiency of this procedure can be dramatically increased by a technique called backtracking (depth-first search with "sanity checks").
Backtracking in Decision Trees

**Idea:** Start at the root of the decision tree and move downwards, that is, **make a sequence of decisions**, until you either reach a solution or you enter a situation from where no solution can be reached by any further sequence of decisions.

In the latter case, **backtrack to the parent** of the current node and take a different path downwards from there. If all paths from this node have already been explored, backtrack to its parent.

Continue this procedure until you **find a solution** or establish that **no solution exists** (there are no more paths to try out).