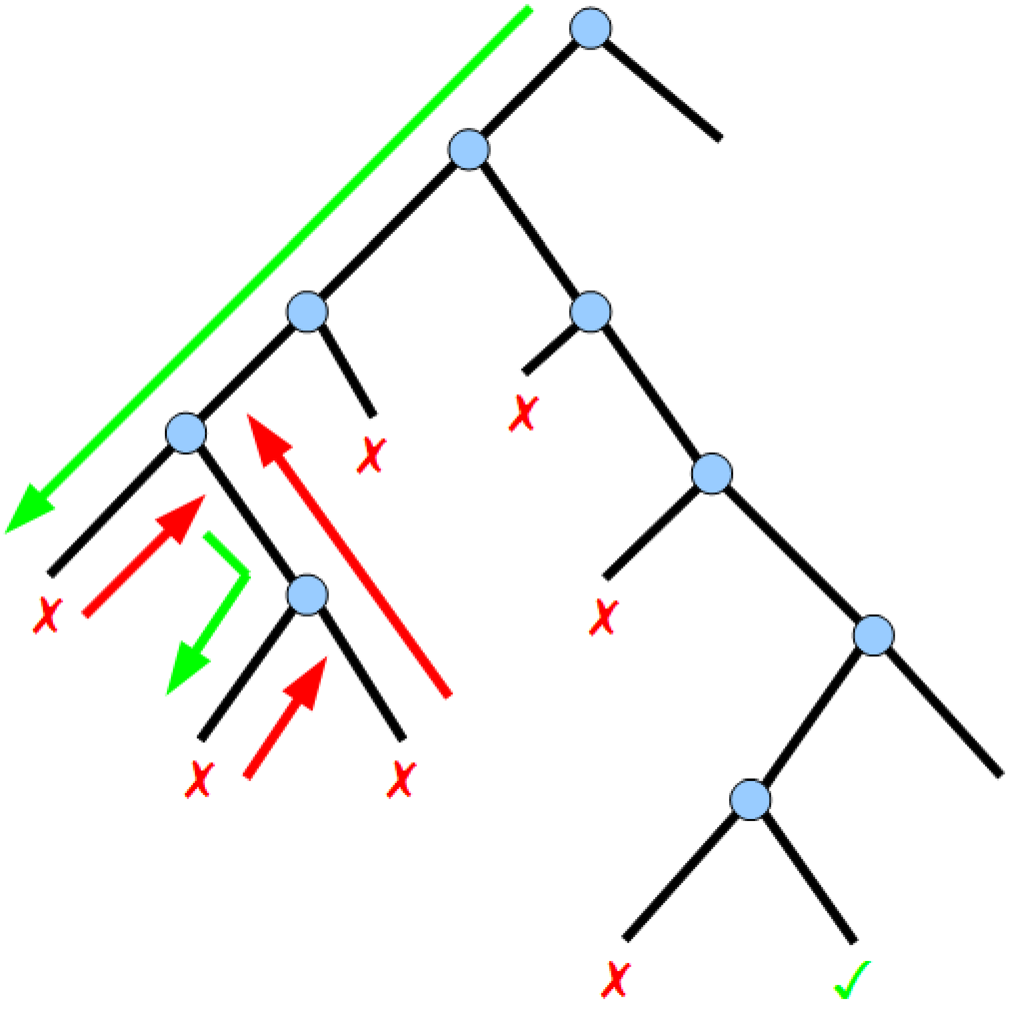
DSE 2256 DESIGN & ANALYSIS OF ALGORITHMS

Lecture 41

Coping with the Limitations of Algorithm Power using Backtracking

n-Queen's Problem, Hamiltonian Circuit Problem, Subset-Sum Problem



Backtracking

Constructs candidate solutions one component at a time and evaluate the **partially constructed solutions** as follows:

- If a partially constructed solution **can be developed further** without violating the problem's constraints it is done by **taking the first remaining legitimate option** for the next component.
- If there is **no legitimate option** for the next component, **no alternatives** for *any* remaining component **need to be considered**.
  - In this case**, the algorithm backtracks** to replace the last component of the partially constructed solution with its next option.

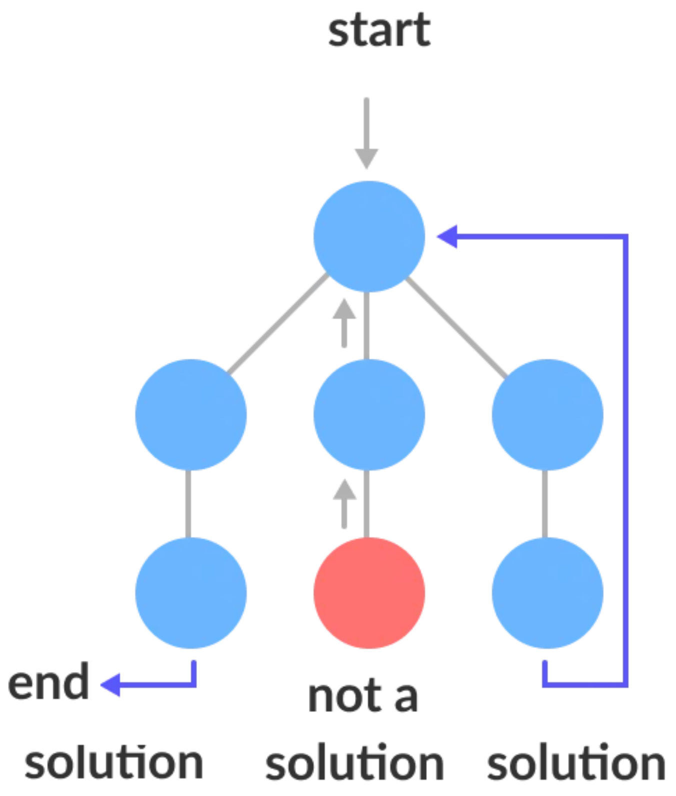
Unlike the Greedy method, once a decision is made it can be revoked (by backtracking).

Backtracking

- To solve any problem using backtracking a **state-space tree** is constructed.

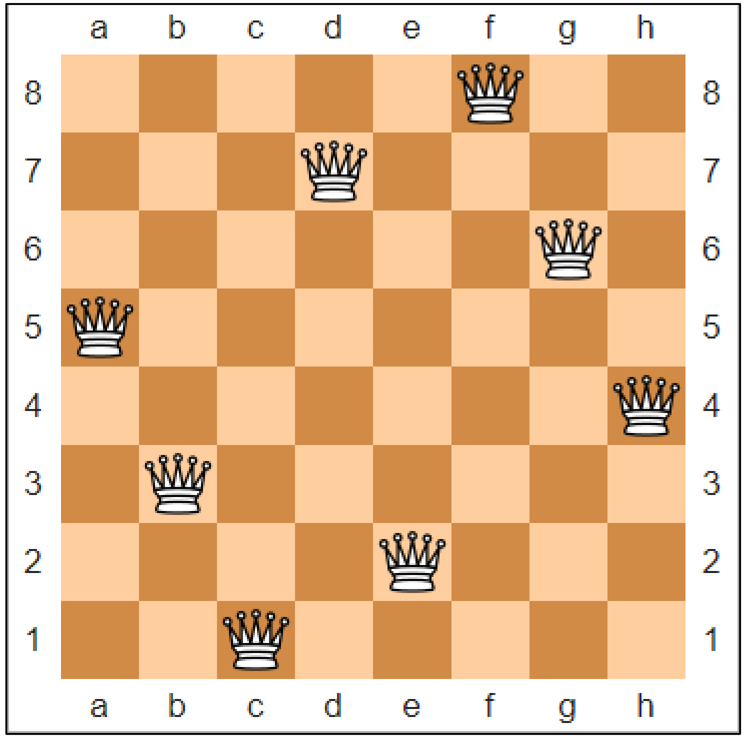
- The state-space tree is constructed based on the **DFS traversal.**

- A node in a state-space tree is said to be :
  - **Promising:** if it corresponds to a partially constructed solution that **may still lead** to a complete solution.
  - **Non-promising:** if it corresponds to a partially constructed solution that **may not lead** to solution.



n-Queen's Problem

Problem: Place **n queens** on an **n × n chessboard** so that no two queens attack each other by being in the **same row** or in the **same column** or on the **same diagonal**.



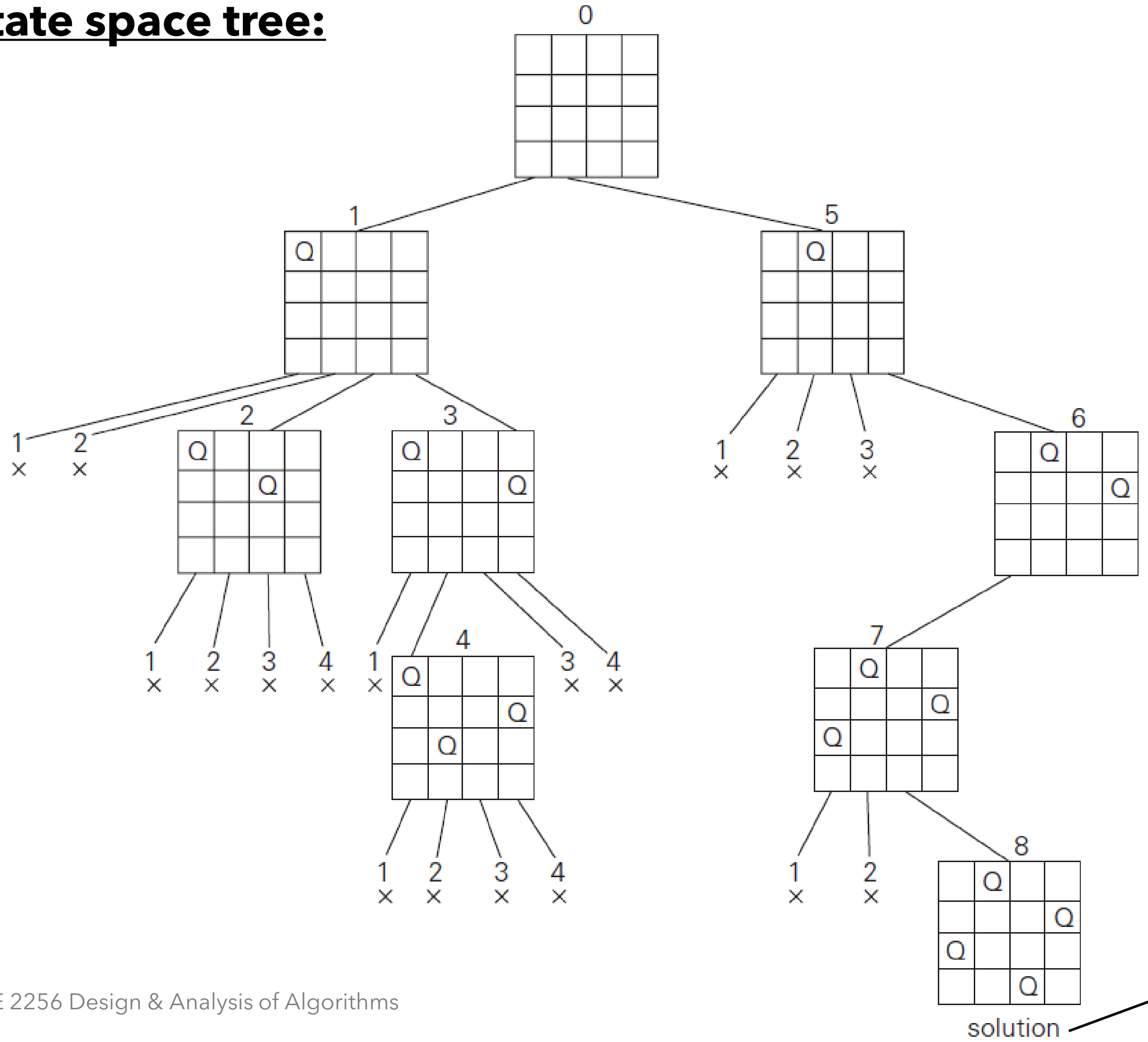
One of the solutions to the 8-Queen's Problem

In 2021, Michael Simkin, a postdoctoral fellow at the Center of Mathematical Sciences and Applications, Harvard University calculated that there are about $\mathbf{(0.143}\boldsymbol{n}\mathbf{)}^{\boldsymbol{n}}$ ways the queens can be placed so none are attacking each other on giant *n*-by *n* chessboards (for larger values of n).

Paper Link: https://arxiv.org/abs/2107.13460

4-Queen's Problem using Backtracking

State space tree:



Time Complexity:

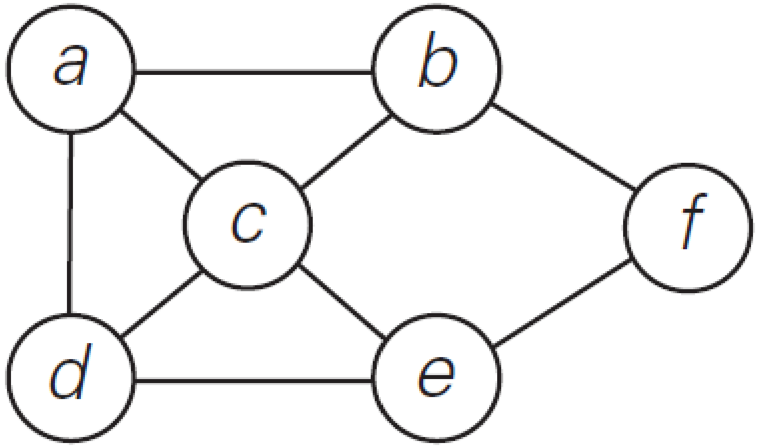
The first queen can be placed in **n** ways, the second queen can be placed in **n-1** ways and so on.

Therefore, the complexity is **O(n!)**

To find all solutions, keep backtracking (from a solution state or from a non-solution state) until all combinations are tried out.

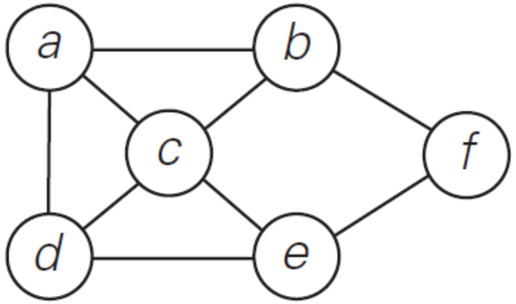
Hamiltonian Circuit Problem

Problem: Finding a Hamiltonian Circuit in a given graph.

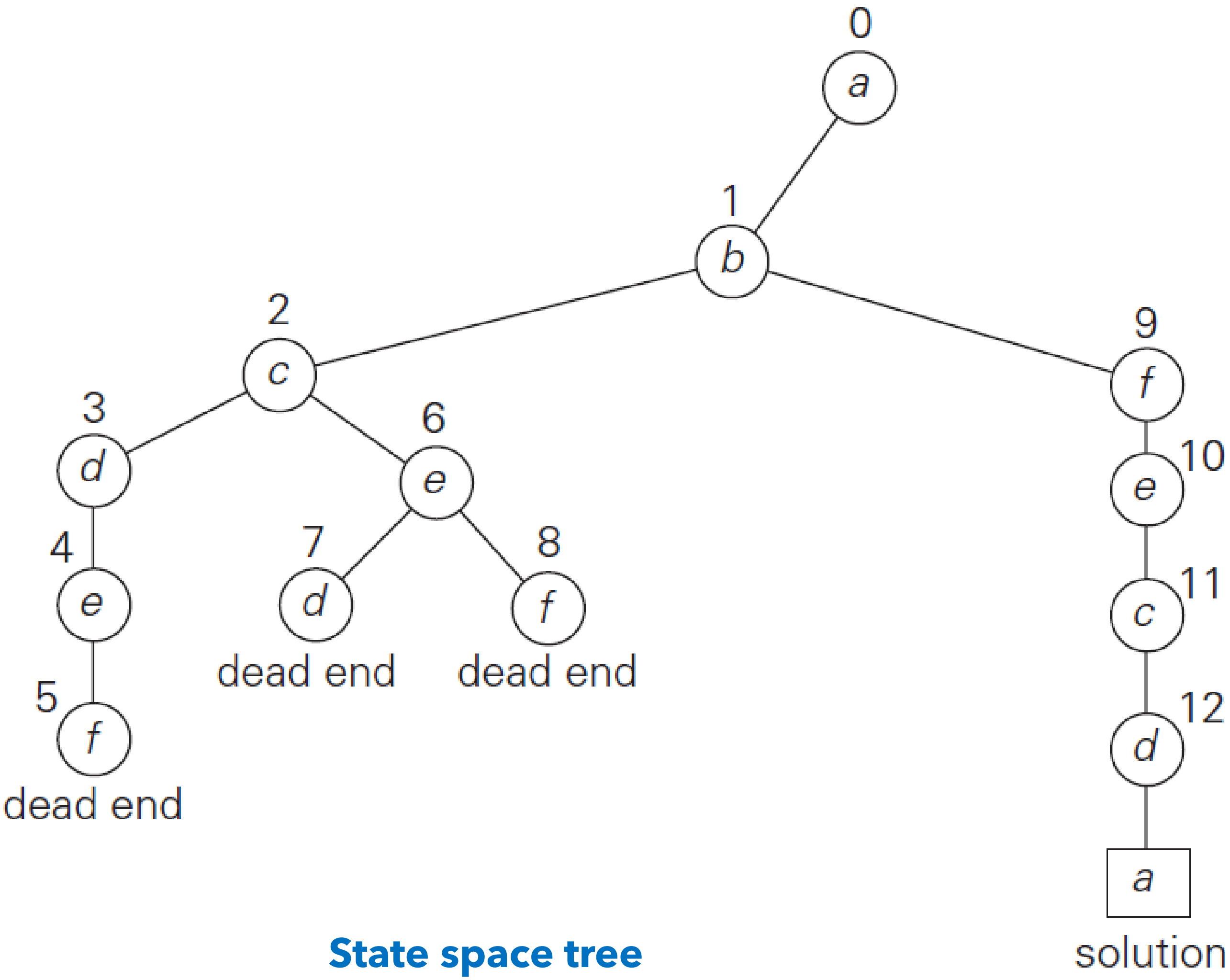


Question: Find the Hamiltonian Circuits for the above graph, if the start vertex is "a" ?

Hamiltonian Circuit Problem using Backtracking



Input Graph



Similarly, other solutions can be found

State space tree

Time Complexity: **O(n!)**

Subset-Sum Problem

Problem: Find a **subset** of a given set **$S = \{s_1, \ldots s_n\}$** of n positive integers whose **sum is equal to** a given positive integer **d**.

<u>Example:</u>

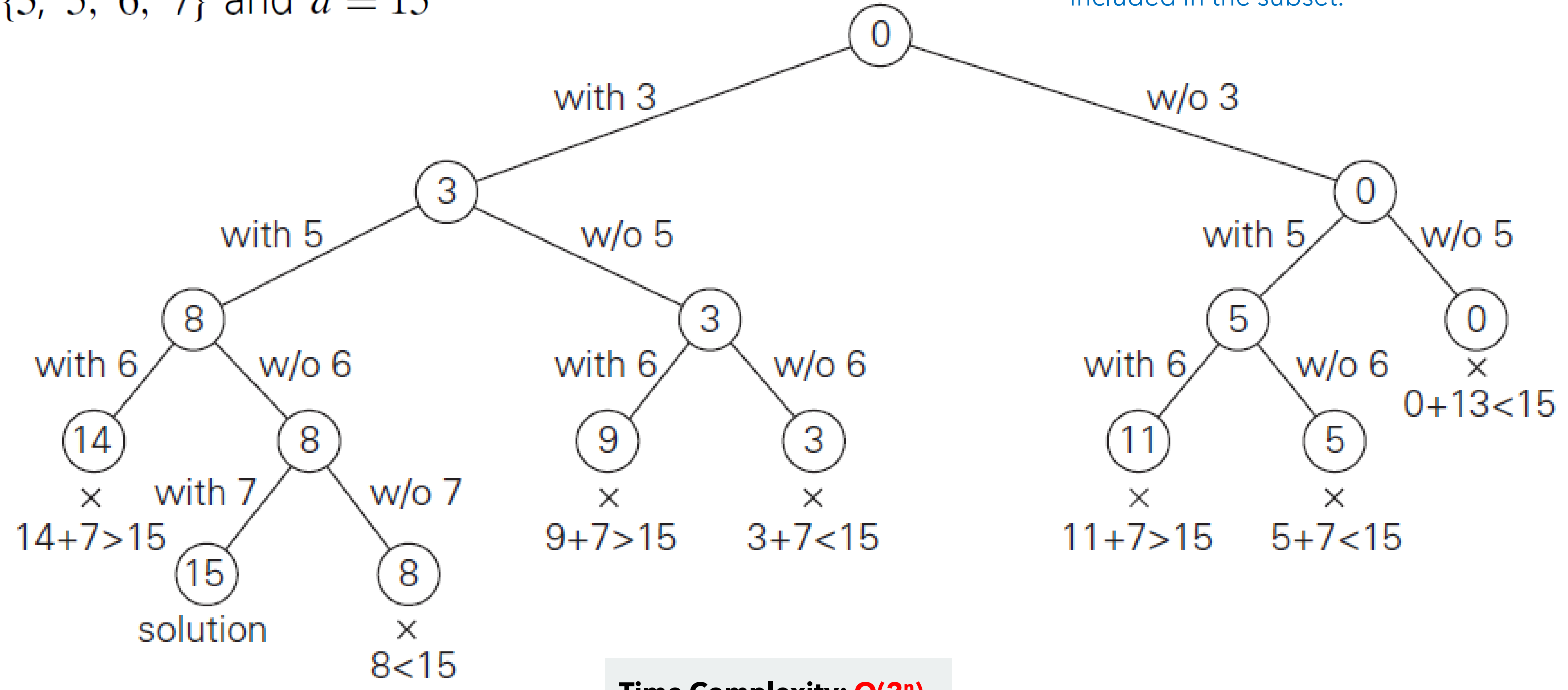
Given S= {1, 2 , 5 , 6 , 8} and d =9 , what are the subsets of S with sum =9 ?

Ans: {1,8} and {1,2, 6}

Subset-Sum Problem using Backtracking

$A = \{3,\ 5,\ 6,\ 7\}$ and $d = 15$

This is the initial state, where no items are included in the subset.



Time Complexity: O(2ⁿ)

Thank you!

Any queries?