SAPIENZA
UNIVERSITÀ DI ROMA

Artificial Intelligence

2023/2024 Prof: Sara Bernardini

Lab 2:
Informed Search

Francesco Argenziano
email: argenziano@diag.uniroma1.it

*The slides have been prepared using the textbook material available on the web, and the slides of the previous editions of the course by Prof. Luigia Carlucci Aiello, Prof. Daniele Nardi and Dott. Fabio Previtali.

Recap

Informed Search: we use problem-specific knowledge to guide the search for the solution -> *evaluation function f(n)*

Recap

Informed Search: we use problem-specific knowledge to guide the search for the solution -> *evaluation function f(n)*

(Perfect) Heuristics: a function *h(n)* that assigns to each state the cost for the cheapest path from the state to a goal state.
h(n) = 0 ⇔ *n* is a goal state.

Recap

Informed Search: we use problem-specific knowledge to guide the search for the solution -> *evaluation function f(n)*

(Perfect) Heuristics: a function *h(n)* that assigns to each state the cost for the cheapest path from the state to a goal state.
h(n) = 0 ⇔ *n* is a goal state.

- f(n) = h(n) -> **Greedy Best-First Search**

Recap

Informed Search: we use problem-specific knowledge to guide the search for the solution -> *evaluation function f(n)*

(Perfect) Heuristics: a function *h(n)* that assigns to each state the cost for the cheapest path from the state to a goal state.
h(n) = 0 ⇔ *n* is a goal state.

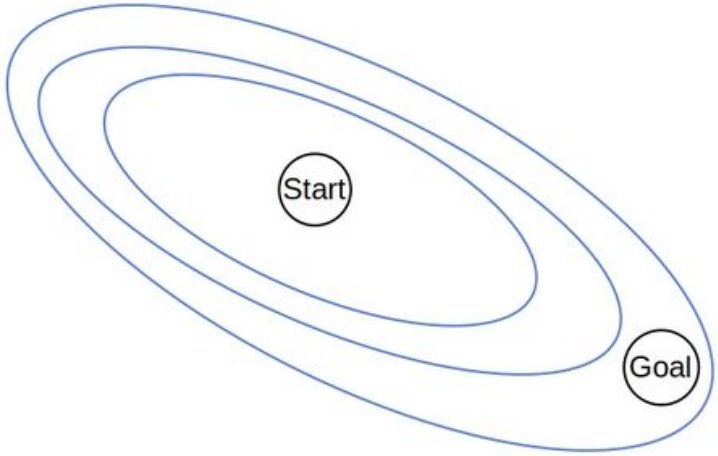
- f(n) = h(n) -> **Greedy Best-First Search**
- f(n) = h(n) + g(n) -> **A***

Properties of A*

- **Optimal? Yes** if the heuristic we use is consistent
- **Complete? Yes** because thanks to g(n) in f(n) we avoid being stuck in a loop

If we assume C* is the cost of optimal solution:

- A* expands **all** nodes with f(n) <= C*
- A* expands **some** nodes with f(n) = C*
- A* expands **no** nodes with f(n) > C*



- **Complexity?** Both exponential in the worst cases

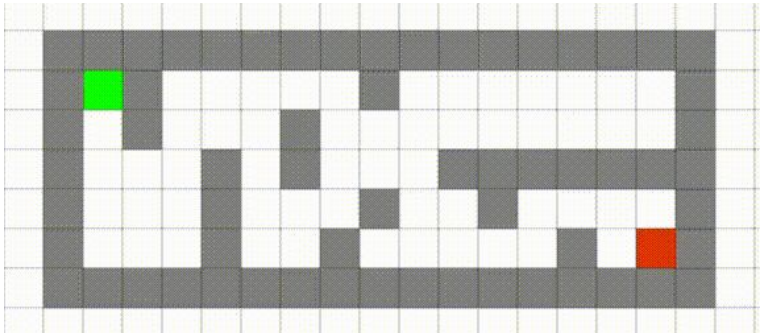
A* vs Dijkstra
https://movingai.com/SAS/ASM/

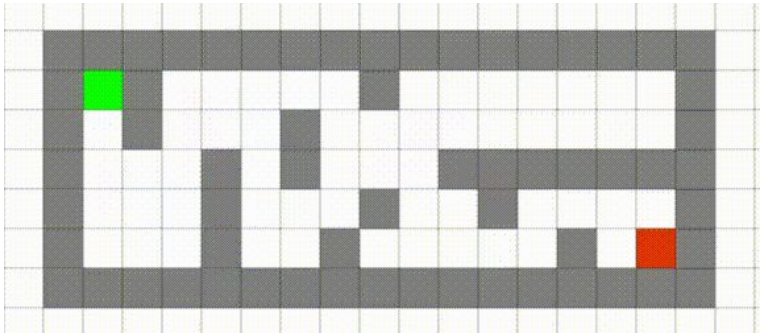
A* breaking ties
https://movingai.com/astar.html

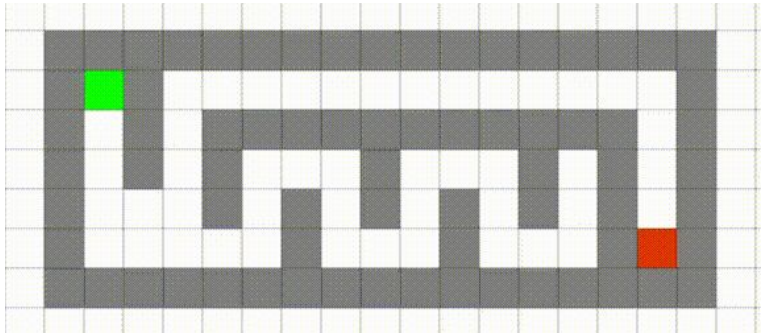
Try yourself!
http://qiao.github.io/PathFinding.js/visual/

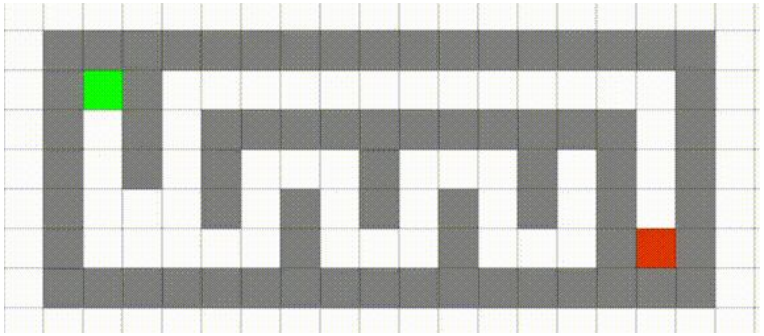
Choice of the heuristics is important!

Manhattan distance vs Euclidean distance







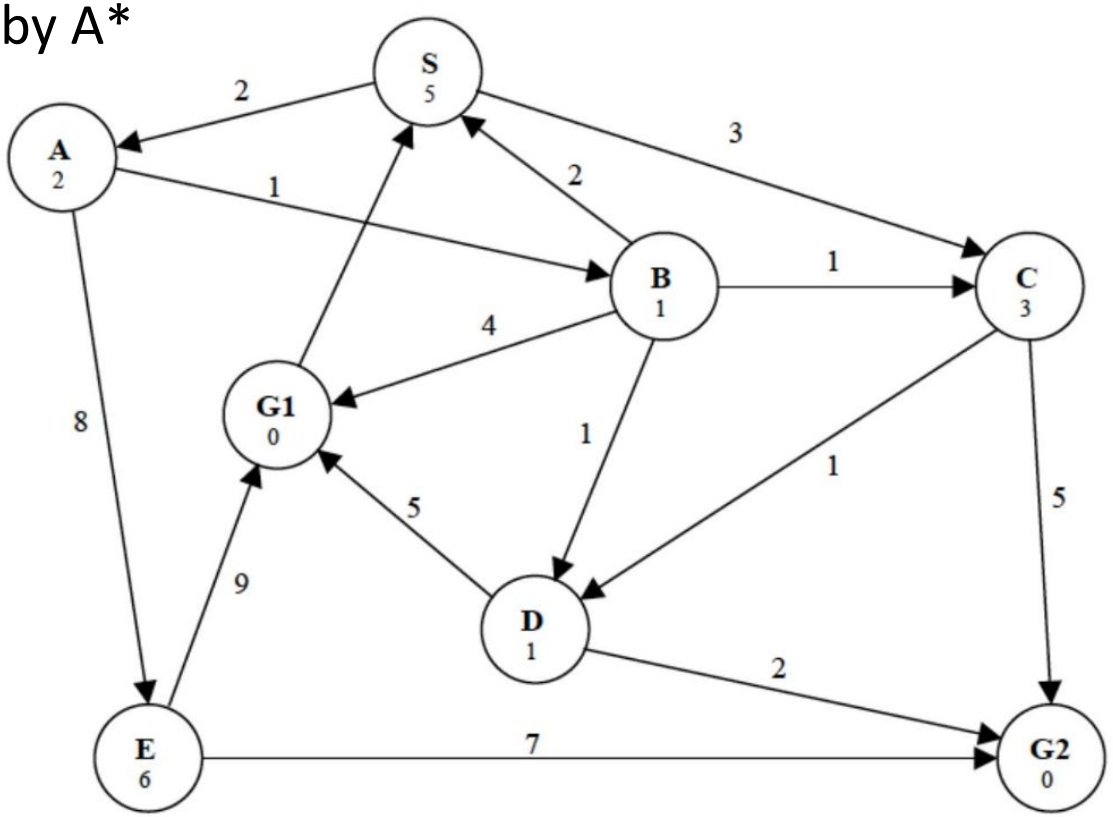


Exercise

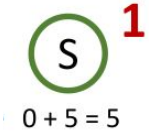
Consider the search space below, where **S** is the start node and **G1** and **G2** satisfy the goal states. Arcs are labelled with the **cost** of traversing them and the **estimated cost** to a goal is reported inside nodes.

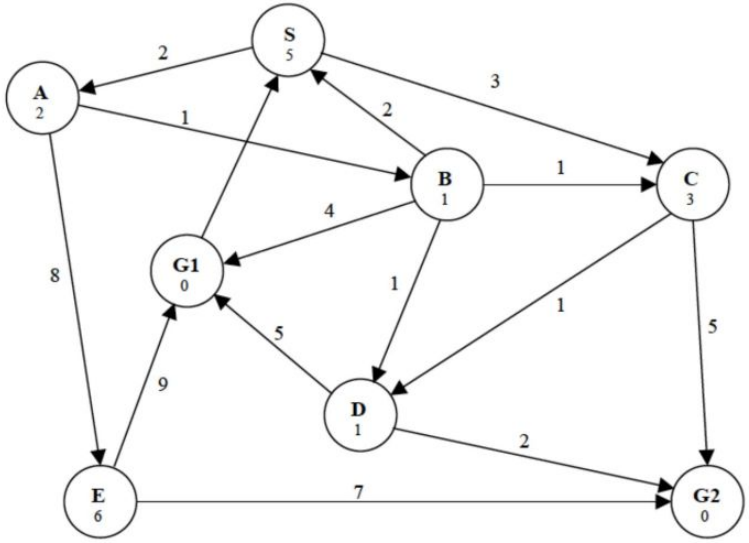
1) Draw the tree generated by A*

2) Mark with increasing natural number the nodes of the tree in the order of their expansion. When all values are equal, nodes are expanded in alphabetical order

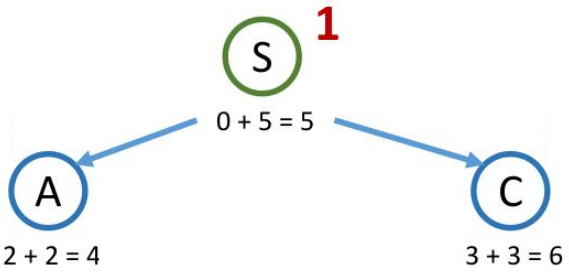


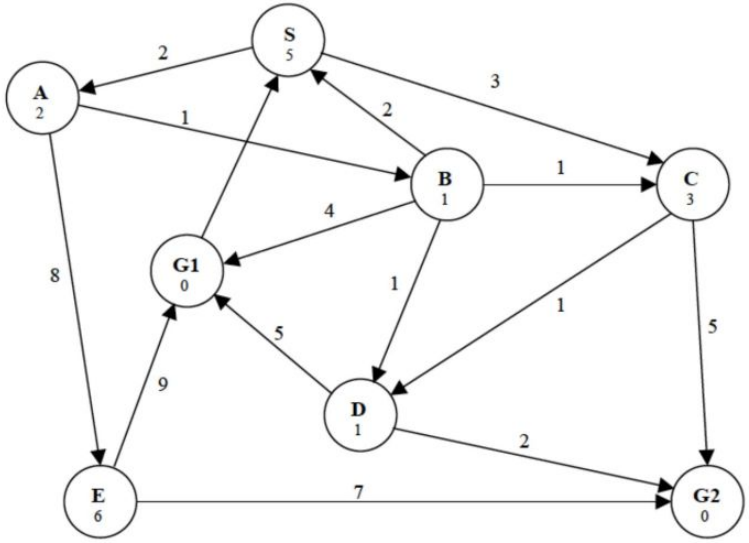
Exercise



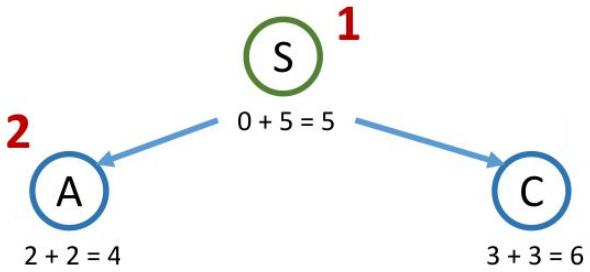


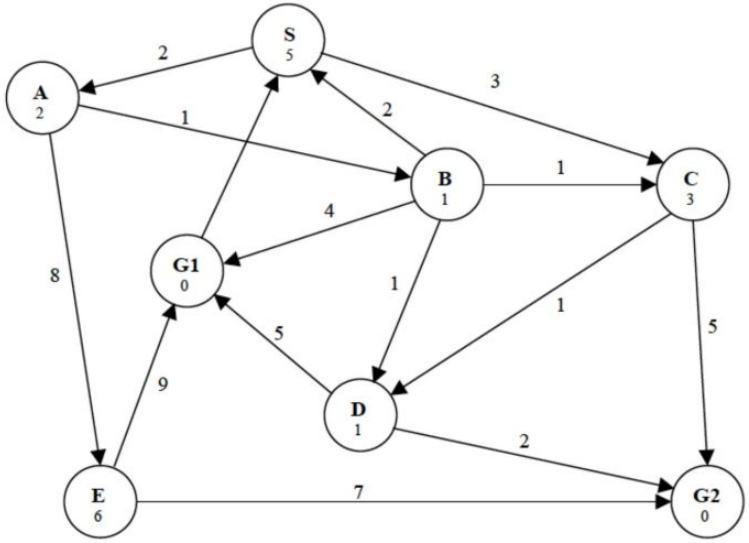
Exercise



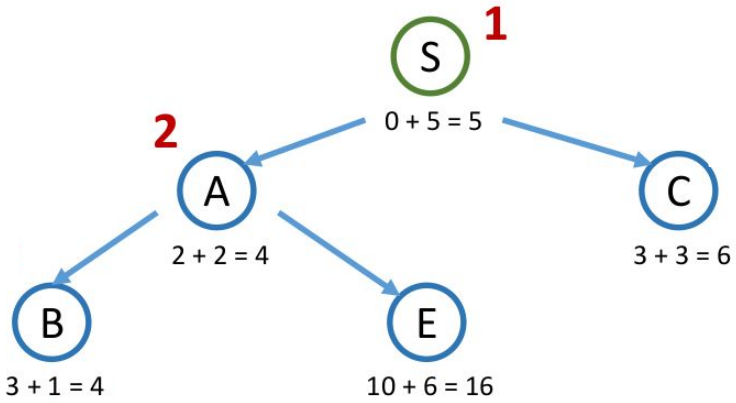


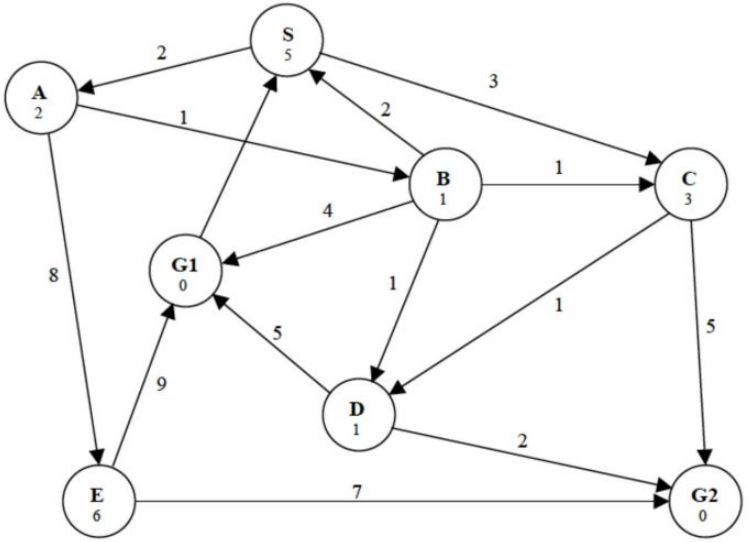
Exercise



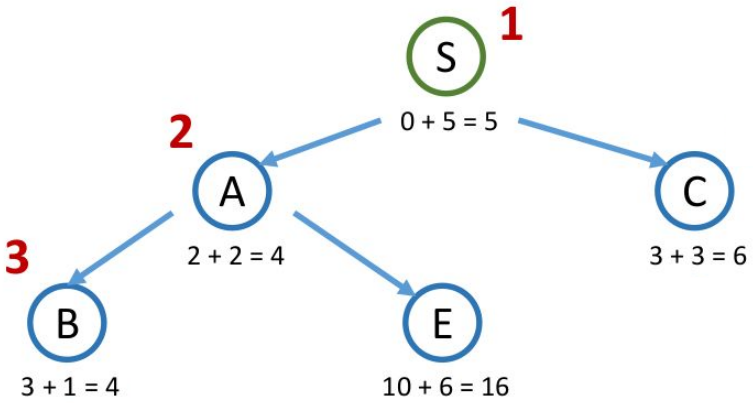


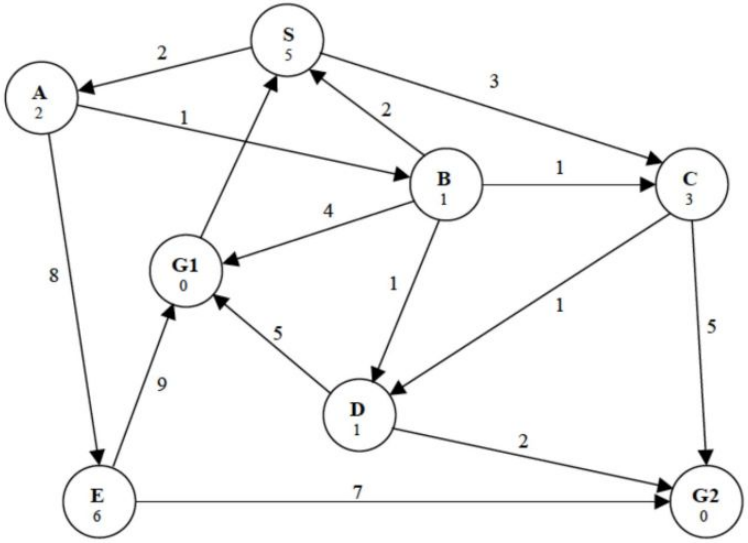
Exercise


1
S
0 + 5 = 5
2
A
2 + 2 = 4
C
3 + 3 = 6
B
3 + 1 = 4
E
10 + 6 = 16

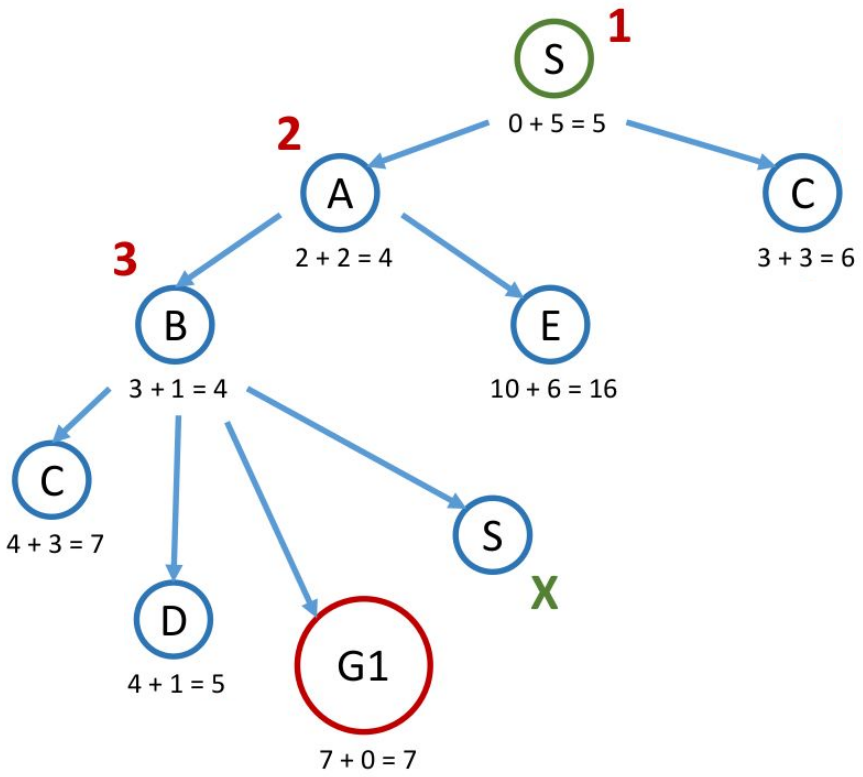

S
5
A
2
B
1
C
3
G1
0
D
1
E
6
G2
0
2
3
1
2
1
4
8
1
5
1
5
9
2
7

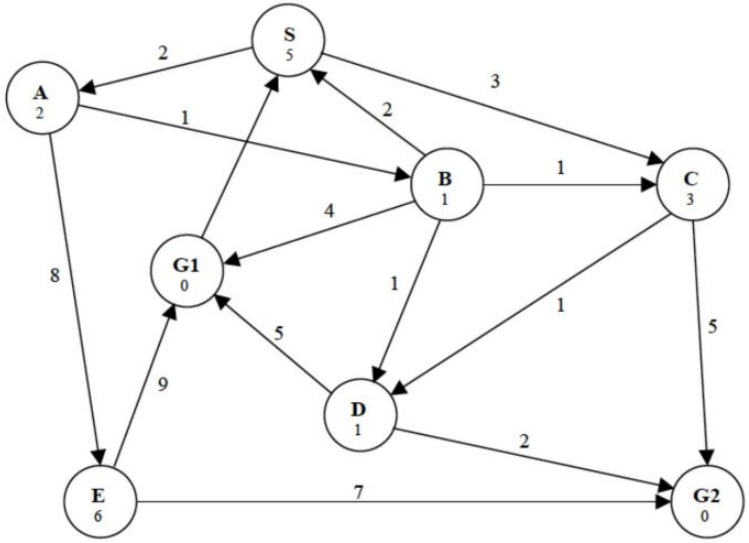
Exercise


1
S
0 + 5 = 5
2
A
2 + 2 = 4
C
3 + 3 = 6
3
B
3 + 1 = 4
E
10 + 6 = 16

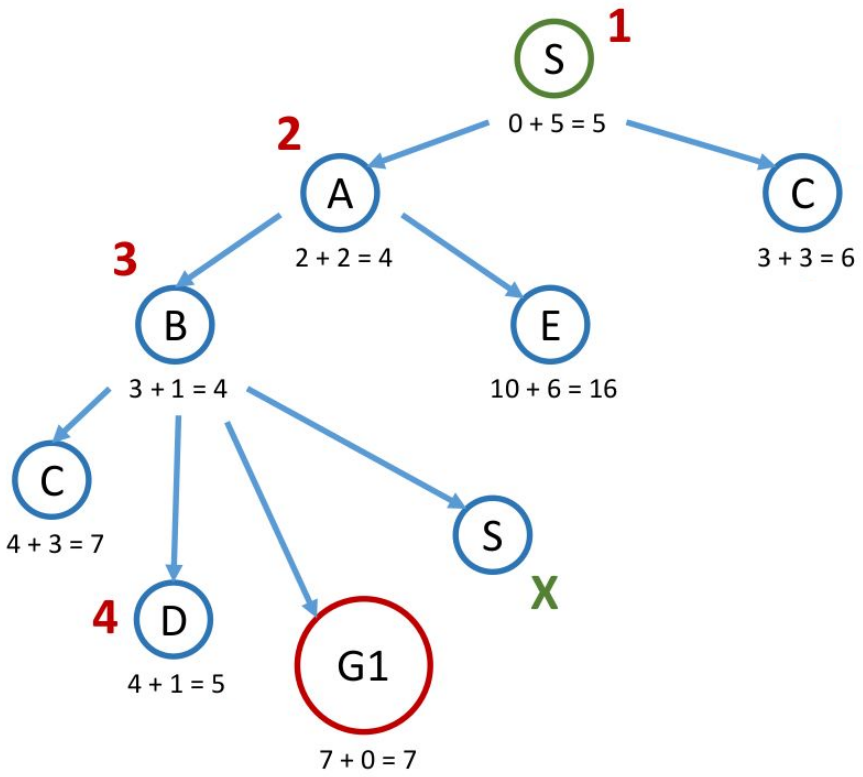

S
5
A
2
B
1
C
3
G1
0
D
1
E
6
G2
0
2
3
1
2
1
4
8
1
1
5
5
9
2
7

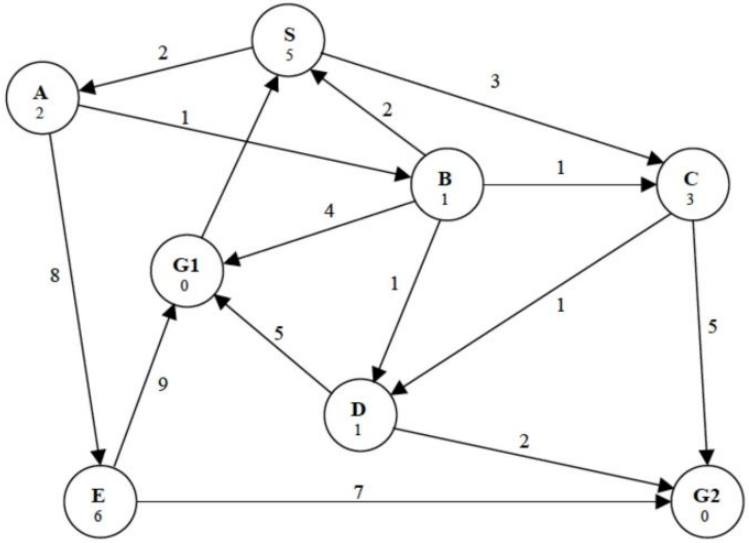
Exercise


1
S
0 + 5 = 5
2
A
2 + 2 = 4
C
3 + 3 = 6
3
B
3 + 1 = 4
E
10 + 6 = 16
C
4 + 3 = 7
D
4 + 1 = 5
G1
7 + 0 = 7
S
X

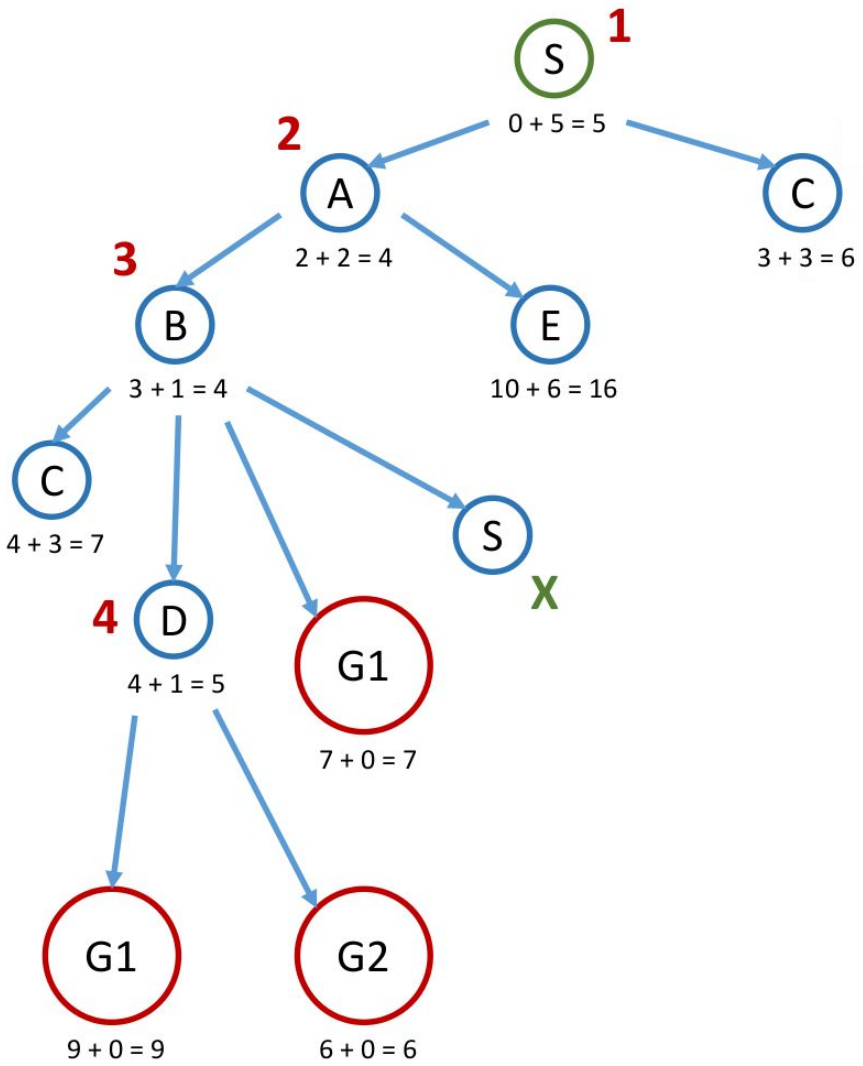

S
5
A
2
B
1
C
3
G1
0
D
1
E
6
G2
0
2
3
1
2
1
4
8
1
1
5
5
9
2
7

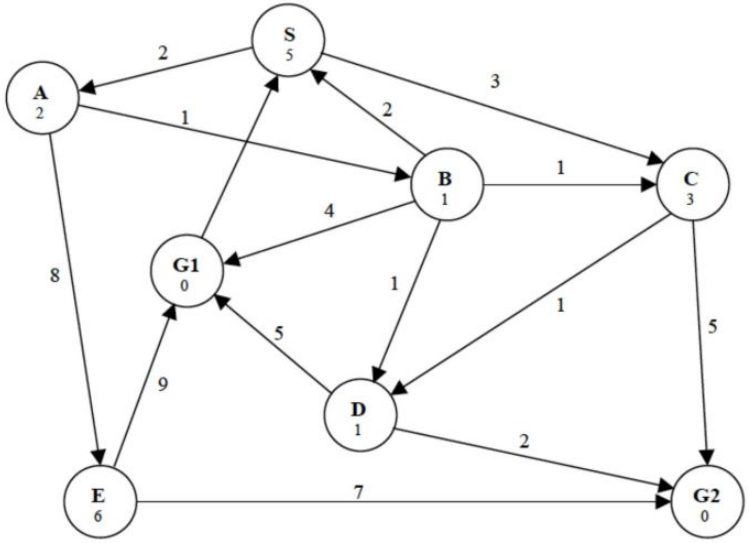
Exercise


S
1
0 + 5 = 5
2
A
2 + 2 = 4
C
3 + 3 = 6
3
B
3 + 1 = 4
E
10 + 6 = 16
C
4 + 3 = 7
S
X
4
D
4 + 1 = 5
G1
7 + 0 = 7

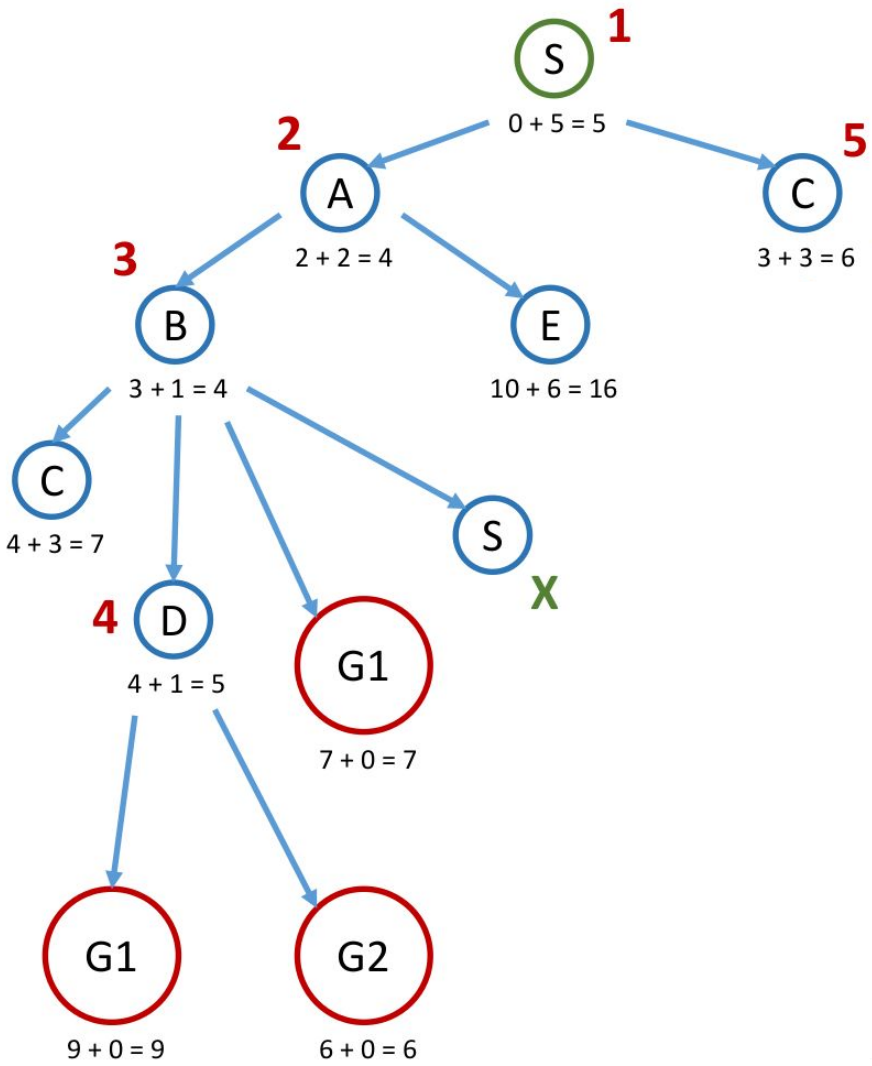

S
5
A
2
B
1
C
3
G1
0
D
1
E
6
G2
0
2
3
1
2
1
4
8
1
1
5
5
9
2
7

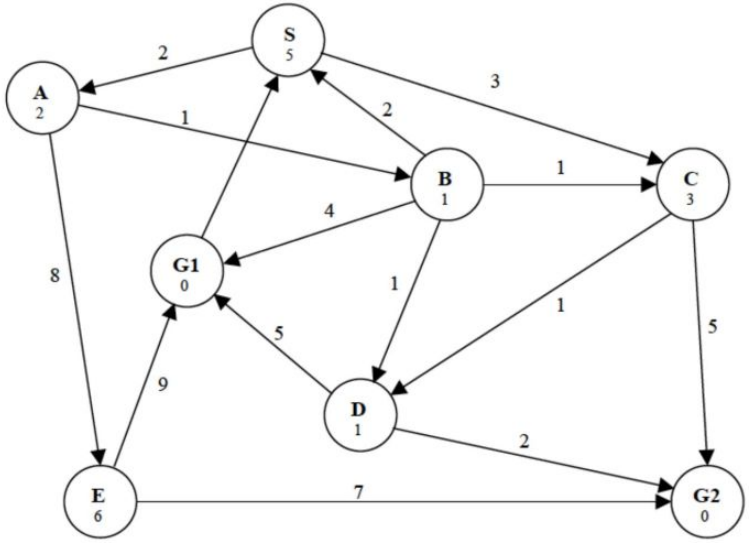
Exercise


1
S
0 + 5 = 5
2
A
2 + 2 = 4
C
3 + 3 = 6
3
B
3 + 1 = 4
E
10 + 6 = 16
C
4 + 3 = 7
S
X
4
D
4 + 1 = 5
G1
7 + 0 = 7
G1
9 + 0 = 9
G2
6 + 0 = 6

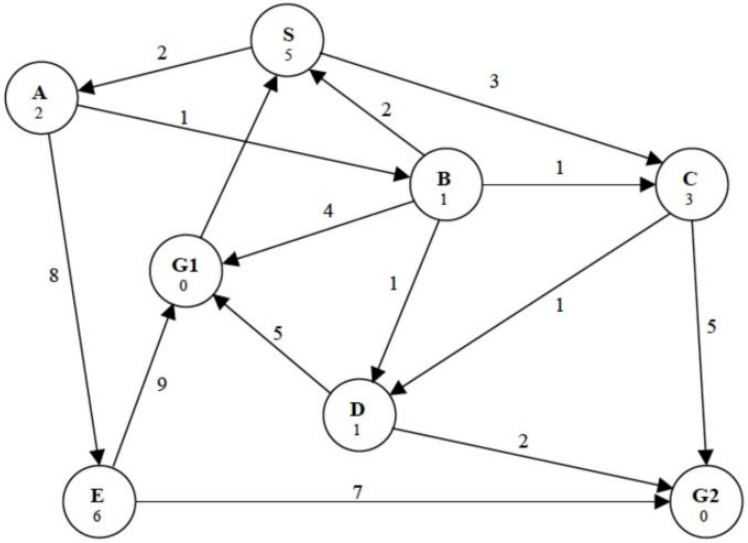

S
5
A
2
B
1
C
3
G1
0
D
1
E
6
G2
0
2
3
1
2
1
4
8
1
1
5
5
9
2
7

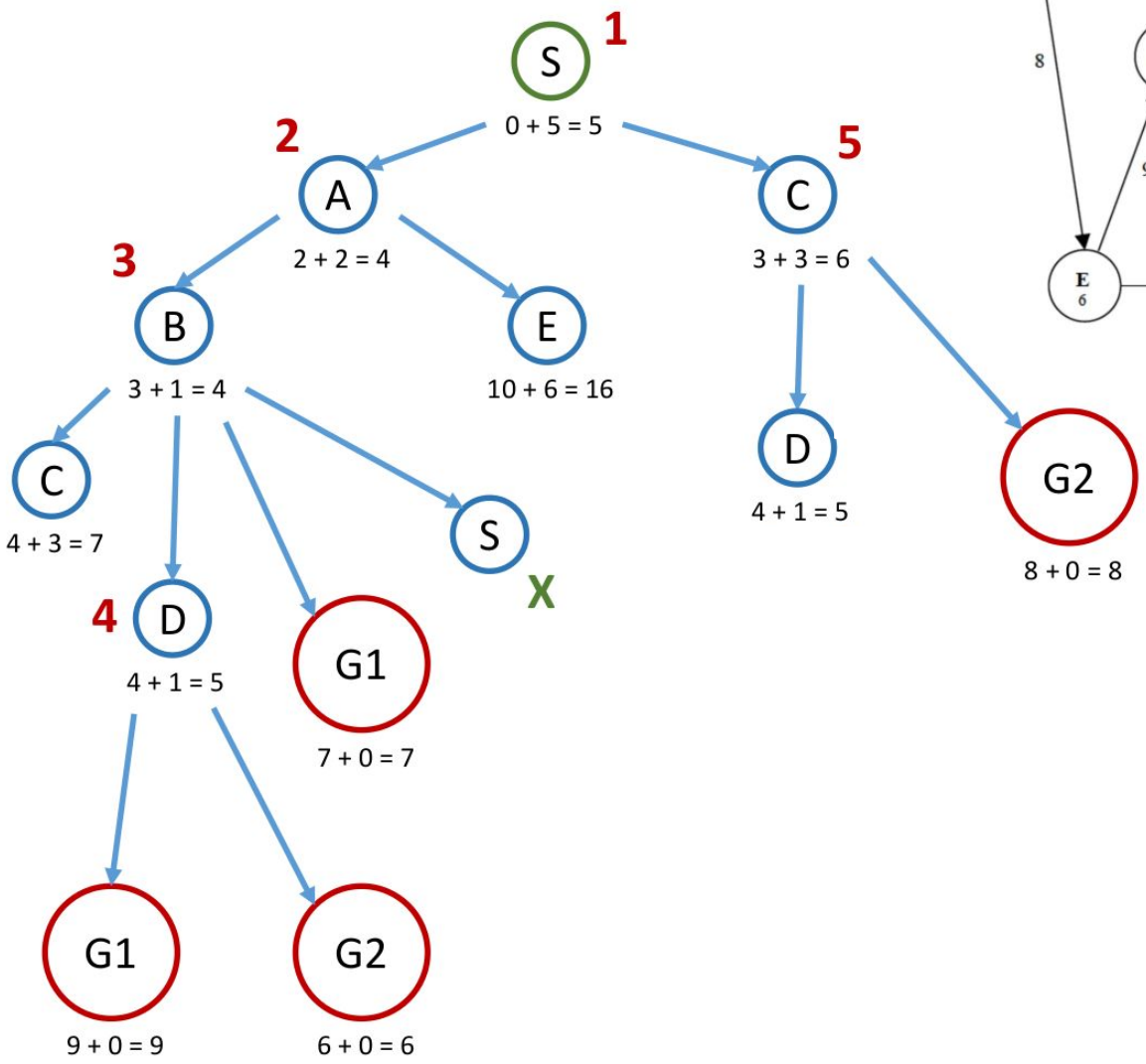
Exercise


S
1
0 + 5 = 5
A
2
2 + 2 = 4
C
5
3 + 3 = 6
B
3
3 + 1 = 4
E
10 + 6 = 16
C
4 + 3 = 7
S
X
D
4
4 + 1 = 5
G1
7 + 0 = 7
G1
9 + 0 = 9
G2
6 + 0 = 6

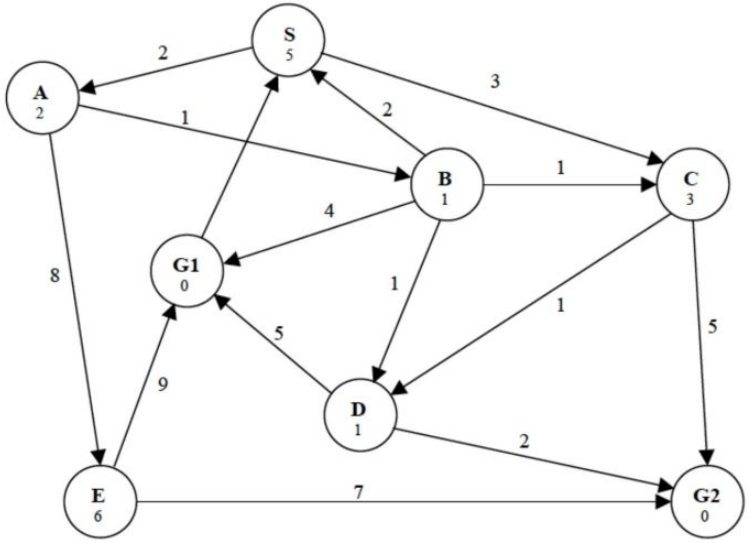

S
5
A
2
B
1
C
3
G1
0
D
1
E
6
G2
0
2
3
1
2
1
4
8
1
1
5
5
9
2
7

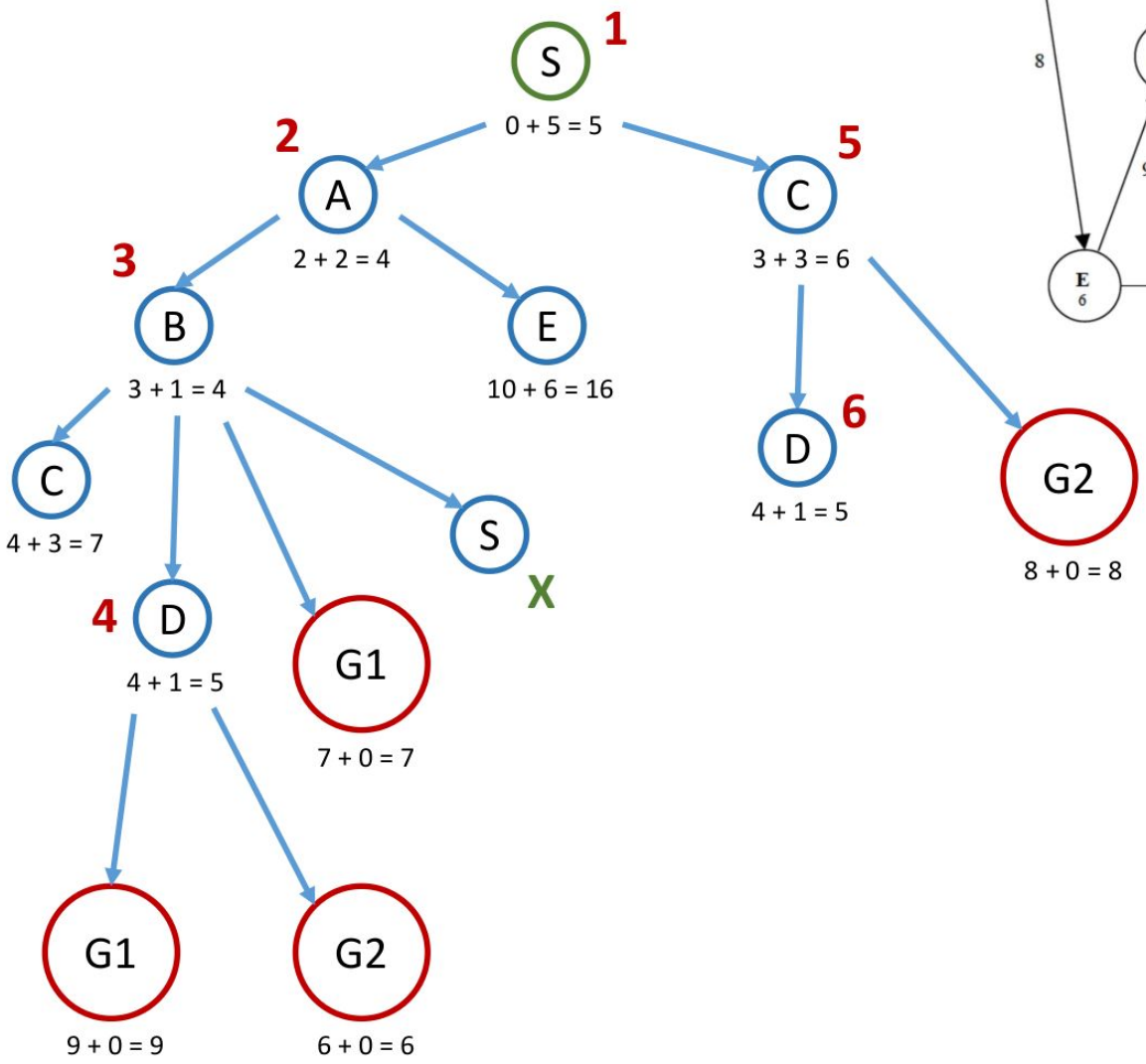
Exercise


S
5
A
2
B
1
C
3
G1
0
D
1
E
6
G2
0
2
3
1
2
1
4
8
1
1
5
5
9
2
7

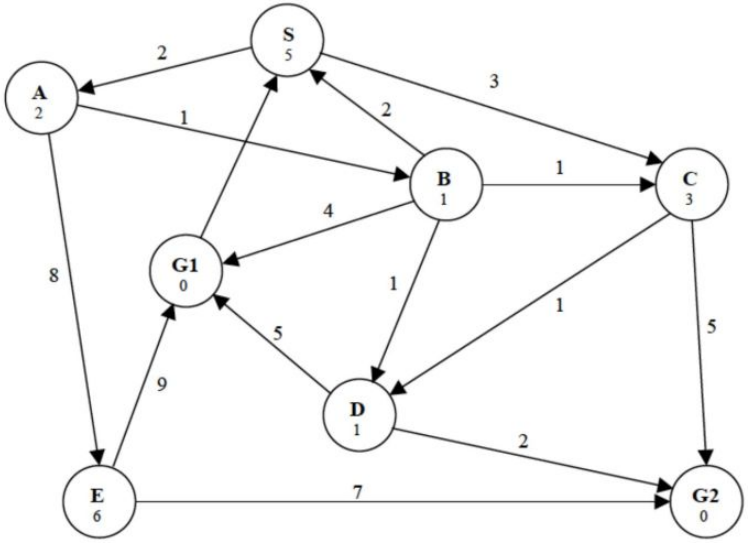

1
S
0 + 5 = 5
2
A
2 + 2 = 4
5
C
3 + 3 = 6
3
B
3 + 1 = 4
E
10 + 6 = 16
D
4 + 1 = 5
G2
8 + 0 = 8
C
4 + 3 = 7
S
X
4
D
4 + 1 = 5
G1
7 + 0 = 7
G1
9 + 0 = 9
G2
6 + 0 = 6

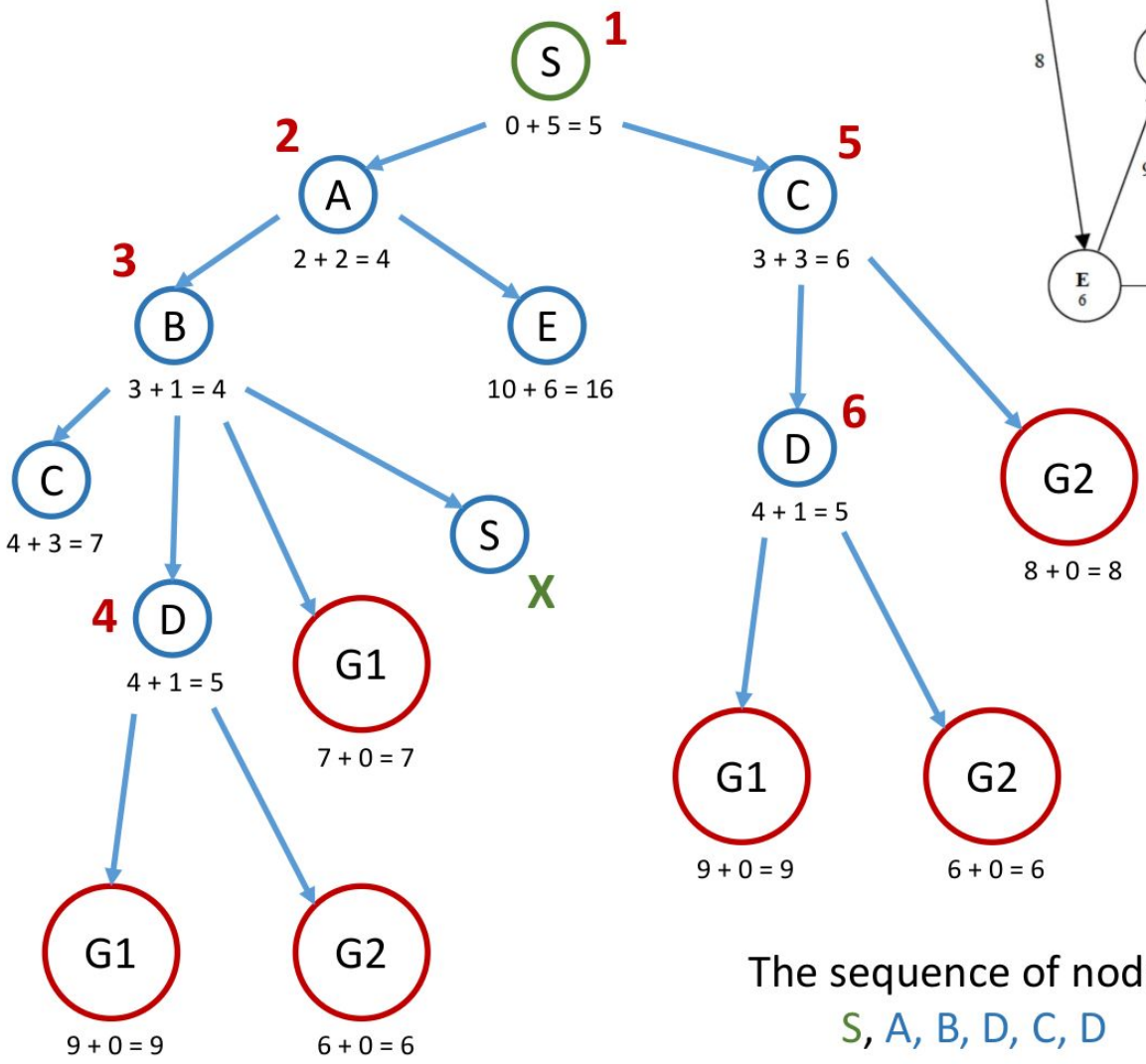
Exercise


S
5
A
2
B
1
C
3
G1
0
D
1
E
6
G2
0
2
3
1
2
1
4
8
1
1
5
5
9
2
7


S
1
0 + 5 = 5
A
2
2 + 2 = 4
C
5
3 + 3 = 6
B
3
3 + 1 = 4
E
10 + 6 = 16
D
6
4 + 1 = 5
G2
8 + 0 = 8
C
4 + 3 = 7
S
X
D
4
4 + 1 = 5
G1
7 + 0 = 7
G1
9 + 0 = 9
G2
6 + 0 = 6

Exercise



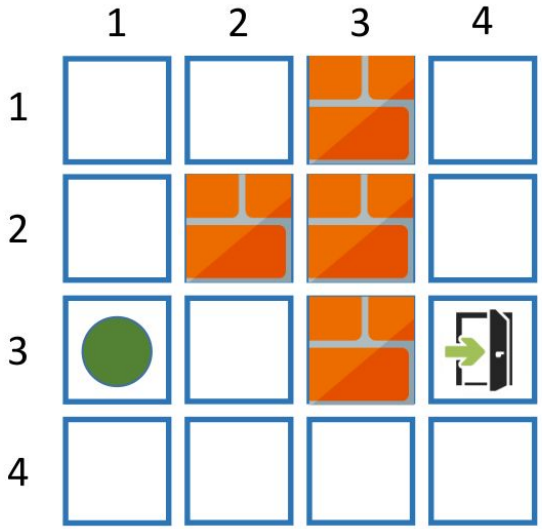


The sequence of nodes expanded by A* is

S, A, B, D, C, D

Exercise: Grid world

An agent is posed at the entrance of the following labyrinth and, it has to traverse it to reach the exit . The symbol represents a wall



The **cost** for going **forward** or **up** is **1**, while for going **down** or **on diagonals** is **2**

The **state space** is the set of possible positions, that can be represented as a pair $\langle i, j \rangle$, with $\mathbf{0 < i < 5}$ and $\mathbf{0 < j < 5}$ and $\langle \boldsymbol{i, j} \rangle \neq$

The **initial** state is in $\langle \mathbf{3, 1} \rangle$ while the **goal** state is in $\langle \mathbf{3, 4} \rangle$

At each step, the agent can move in every direction to one of the adjacent cells. It can perform an horizontal, vertical and diagonal move and, it can only advance from left to right, i.e. it cannot go from $\langle i, j \rangle$ to $\langle i, j-1 \rangle$, $\langle i-1, j-1 \rangle$ not $\langle i+1, j-1 \rangle$. Of course, the agent cannot traverse walls nor move out of the grid

Exercise: Grid world

Operators

Name	Meaning	Effect	Cost
$up(\langle i,j\rangle)$	Go up	$\langle i+1,j\rangle$	1
$d-up(\langle i,j\rangle)$	Go diagonal up	$\langle i+1,j+1\rangle$	2
$forward(\langle i,j\rangle)$	Go forward	$\langle i,j+1\rangle$	1
$down(\langle i,j\rangle)$	Go down	$\langle i-1,j\rangle$	2
$d-down(\langle i,j\rangle)$	Go diagonal down	$\langle i-1,j+1\rangle$	2

Exercise: Grid world

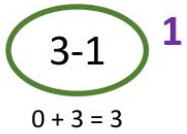
- Is the **Manhattan distance** an appropriate **heuristic** function? Is it admissible?

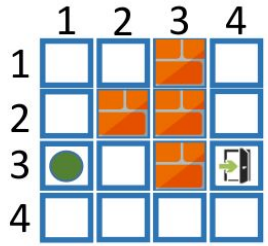
- List the expanded nodes in the **order of expansion**

- Use **A*** with the above heuristic function to find a solution. **List all the generated nodes** with the order of generation and the values for g, h and f. **When more than one node has the same minimal value for f expand the most recently generated one**

Exercise: Grid world

Goal state: 3-4, g(s) = as in the table, h(s) = Manhattan distance



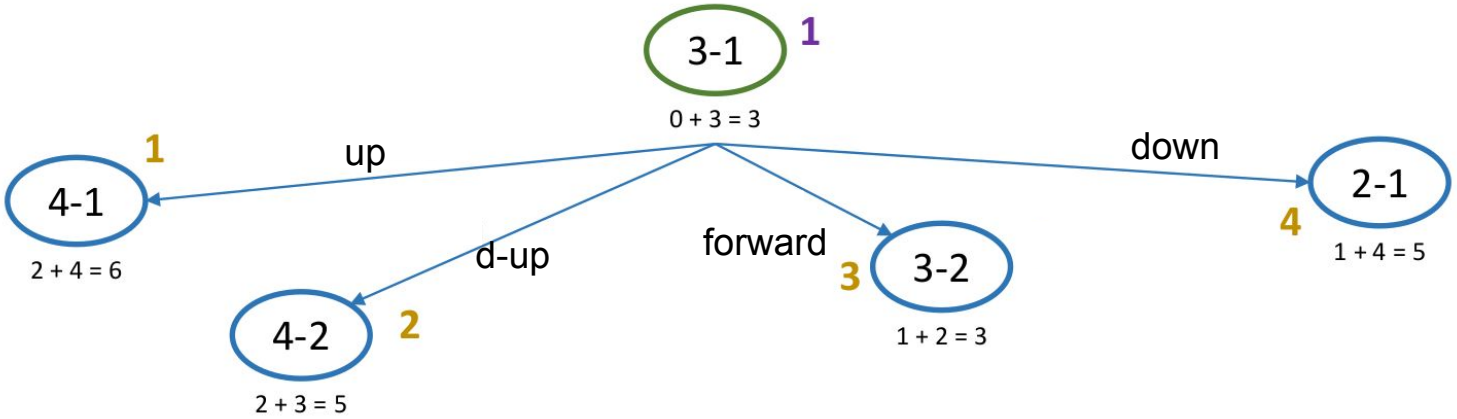


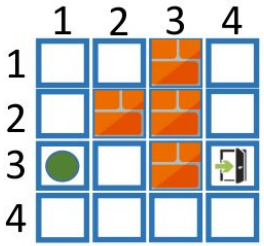
Generation

Expansion

Exercise: Grid world

Goal state: 3-4, g(s) = as in the table, h(s) = Manhattan distance



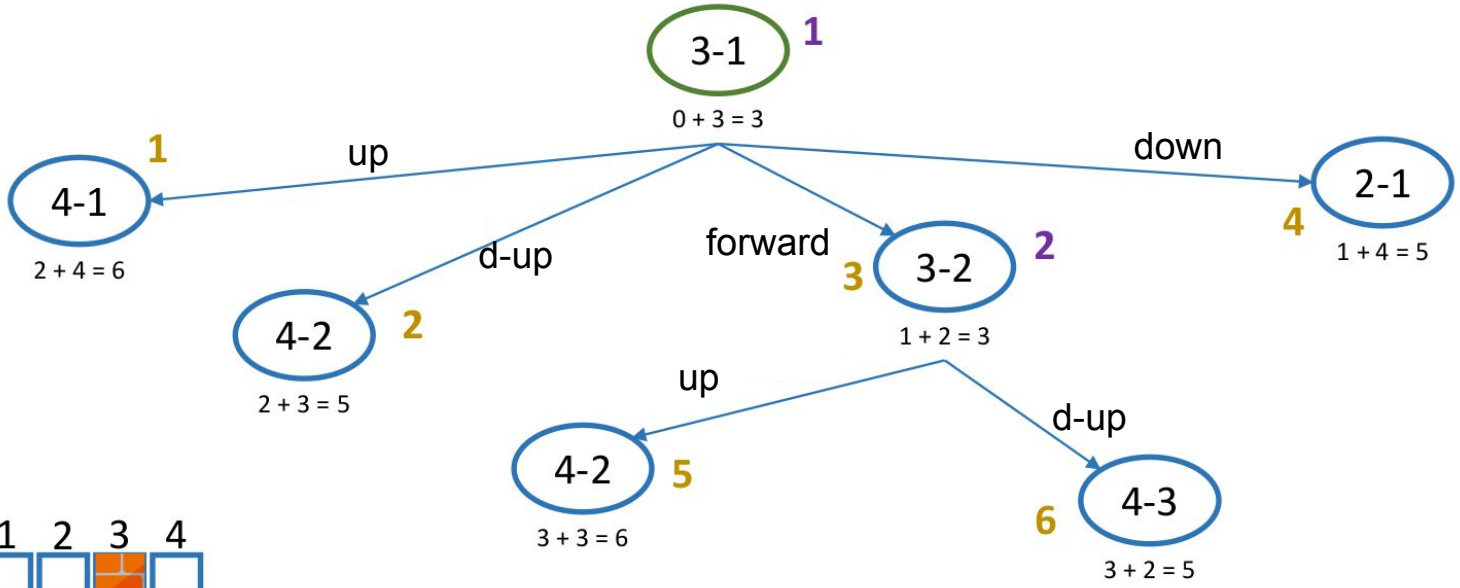


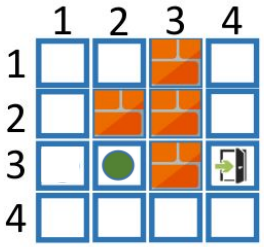
Generation

Expansion

Exercise: Grid world

Goal state: 3-4, g(s) = as in the table, h(s) = Manhattan distance



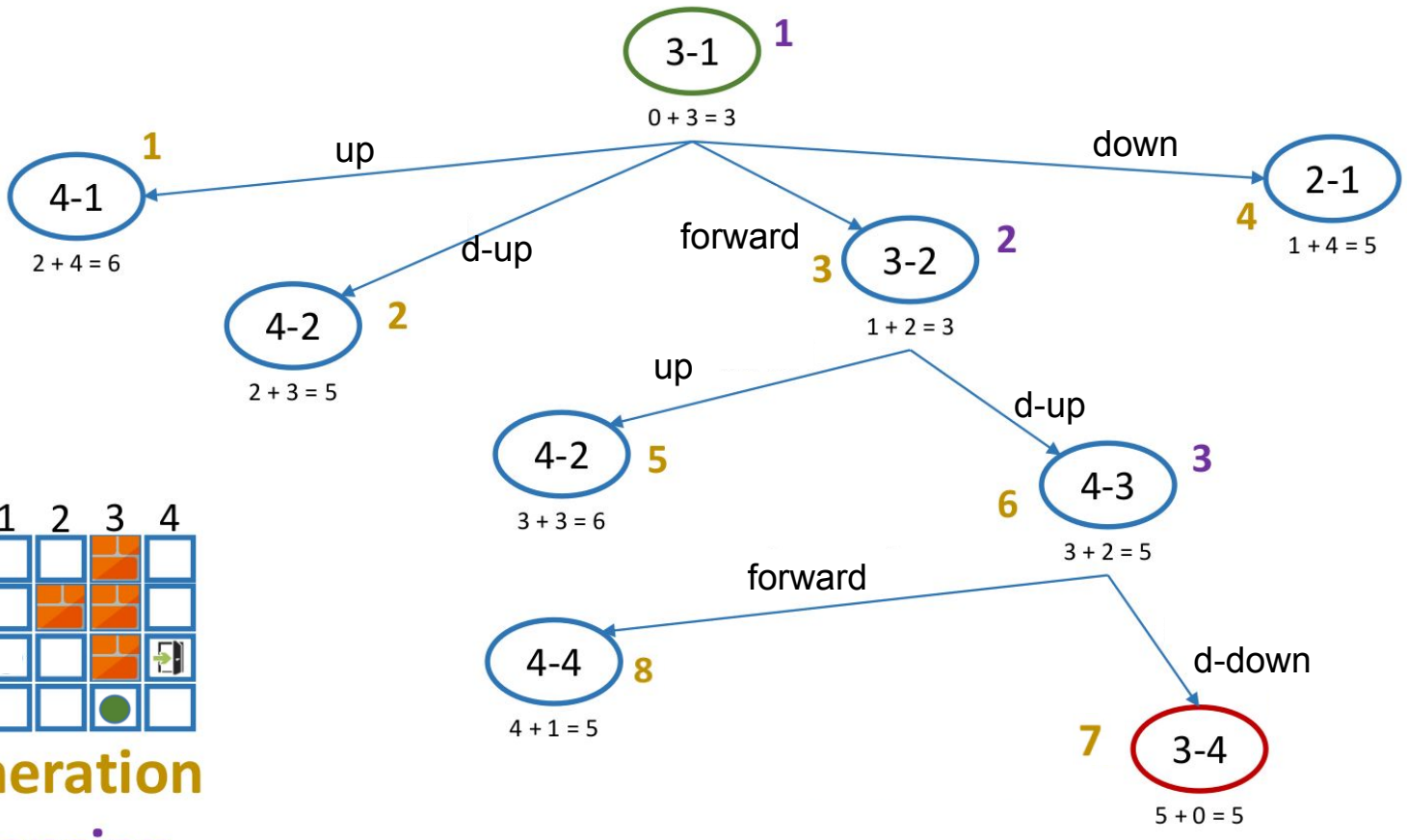


Generation

Expansion

Exercise: Grid world

Goal state: 3-4, g(s) = as in the table, h(s) = Manhattan distance

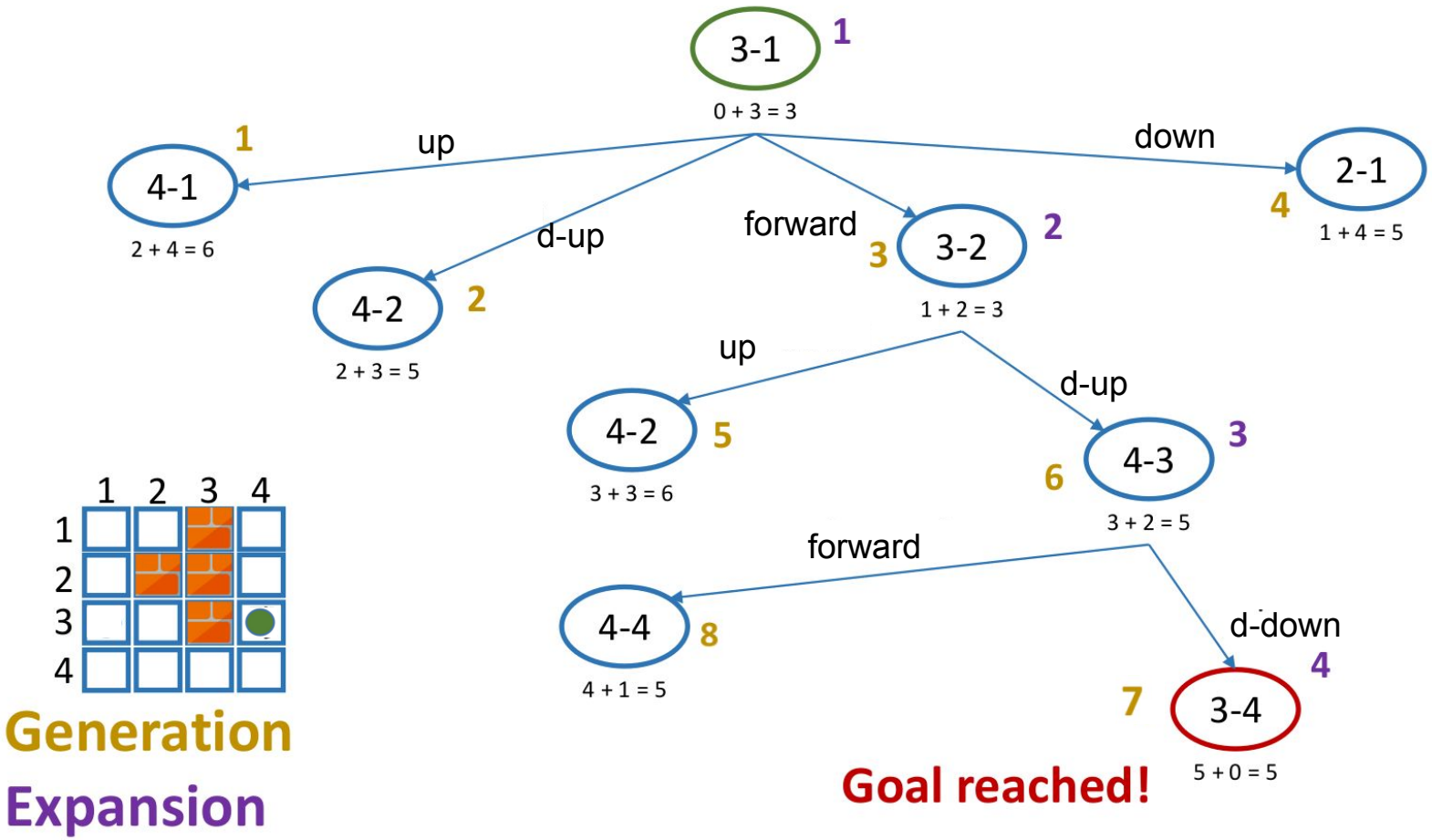


Generation

Expansion

Exercise: Grid world

Goal state: 3-4, g(s) = as in the table, h(s) = Manhattan distance



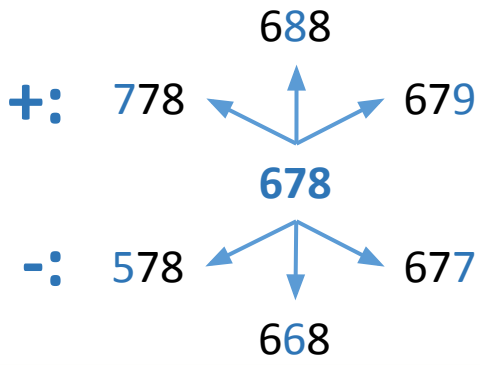
Exercise: Forbidden numbers

- The game is about integer numbers between 100 and 999
- Two numbers are given to express the start and goal situations
- A set of numbers is defined as the set of forbidden numbers (eg. {679, 666})
- A move changes a single digit in the number by adding or subtracting one

Constraints:

- 1 cannot be added to 9 nor subtracted from 0
- Avoid forbidden numbers
- Consecutive changes of the same digit are not allowed:
  677 -> 678 -> 679

eg.



Exercise: Forbidden numbers

1. Characterize the **state space**
2. Specify the **operators**
3. Find a minimal sequence of moves to go from **I** = 567 to **G** = 777.
4. **Forbidden** = {666, 667}
5. Find a good **heuristics** to be used by **A***
6. **Draw** the **search tree** generated by A* to solve the problem.
7. For each node **indicate**: the number (state), f, g, and h and an integer indicating the expansion order

Exercise: Forbidden numbers

$S = \{(N, o) \mid N = num(x,y,z), N \notin F \text{ and } o \in O\}$

$num(x, y, z) = 100*x + 10*y + z$
with $y, z \in \{0,1,..,9\}$ and $x \in \{1,..,9\}$
$O = \{u, t, h, no\text{-}op\}$

$I = (num(5,6,7), no\text{-}op)$
$G = \{(num(7,7,7), o) \mid o \in O\}$

$h(n) = |x-7| + |y-7| + |z-7|$

Exercise: Forbidden numbers

Specify the **operators**

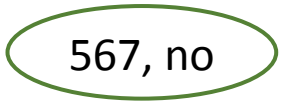
State = (num(x,y,z), o)

Op	Conditions	New state
add(u)	z!=9 and o!=u and num(x,y,z+1) not in F	(num(x,y,z+1), u)
sub(u)	z!=0 and o!=u and num(x,y,z-1) not in F	(num(x,y,z-1), u)
add(t)	y!=9 and o!=t and num(x,y+1,z) not in F	(num(x,y+1,z), t)
sub(t)	y!=0 and o!=t and num(x,y-1,z) not in F	(num(x,y-1,z), t)
add(h)	x!=9 and o!=h and num(x+1,y,z) not in F	(num(x+1,y,z), h)
sub(h)	o!=h and num(x-1,y,z) not in F	(num(x-1,y,z), h)

Exercise: Forbidden numbers

A* tree

G=777, Forbidden set: {666, 667}

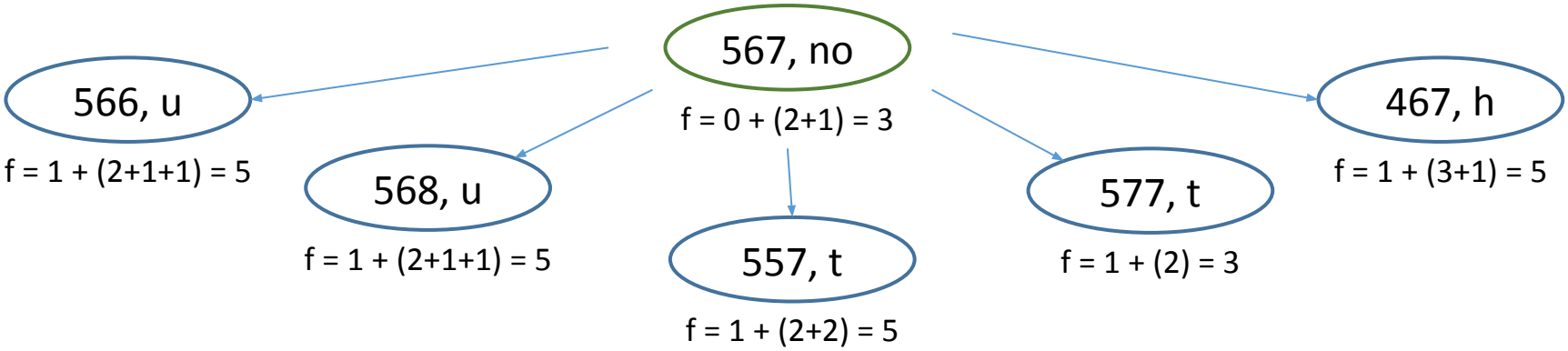


f = 0 + (2+1) = 3

Exercise: Forbidden numbers

A* tree

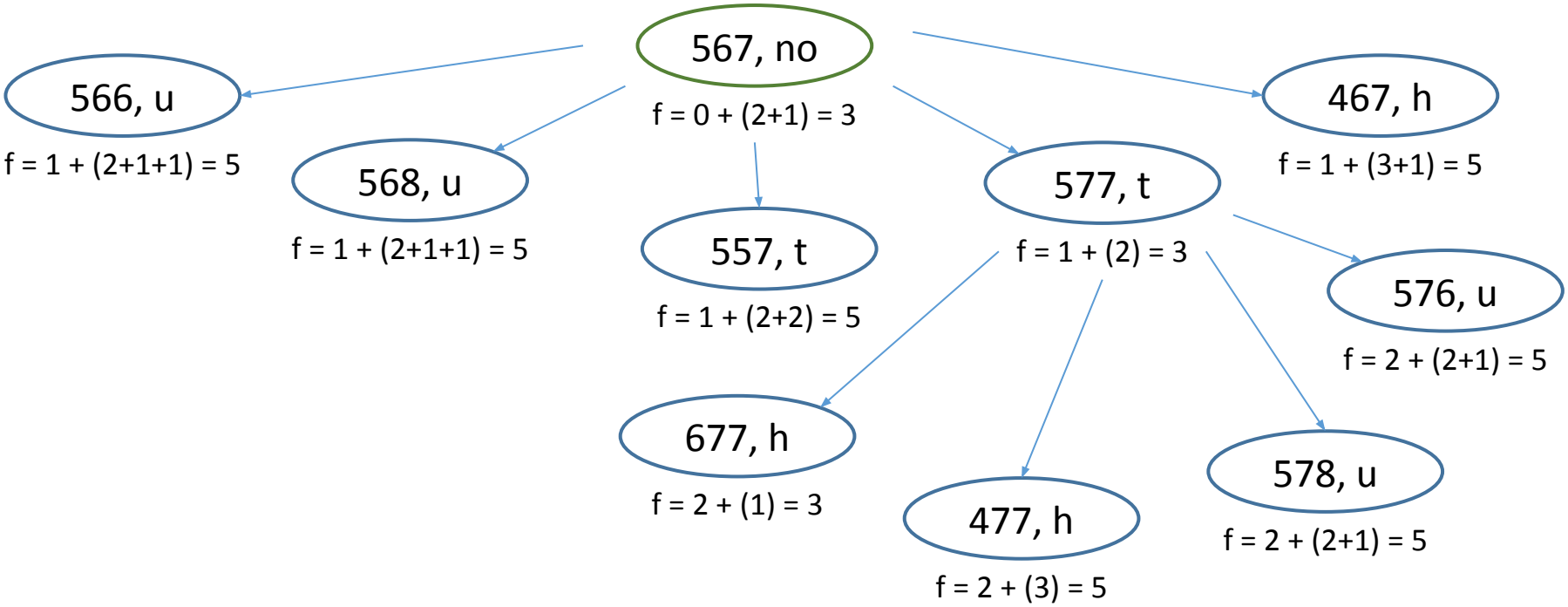
G=777, Forbidden set: {666, 667}



Exercise: Forbidden numbers

A* tree

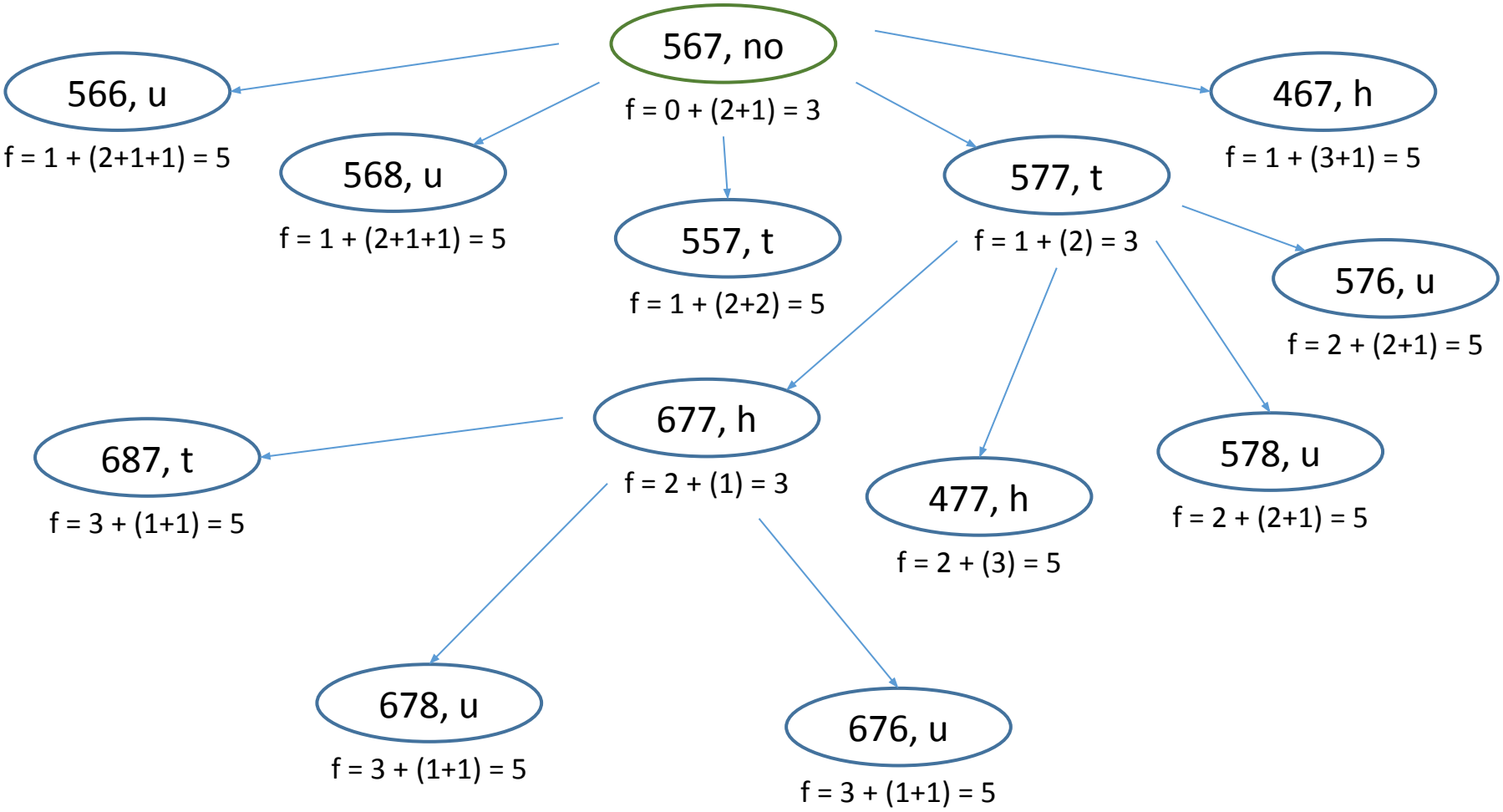
G=777, Forbidden set: {666, 667}



Exercise: Forbidden numbers

A* tree

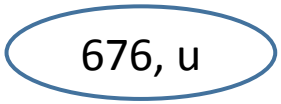
G=777, Forbidden set: {666, 667}



Exercise: Forbidden numbers

A* tree

G=777, Forbidden set: {666, 667}

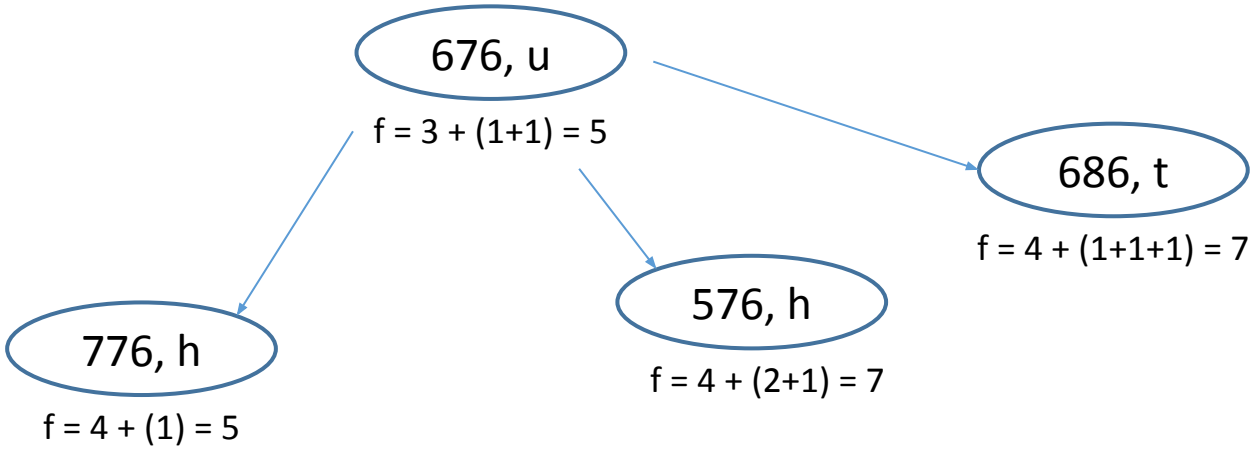


f = 3 + (1+1) = 5

Exercise: Forbidden numbers

A* tree

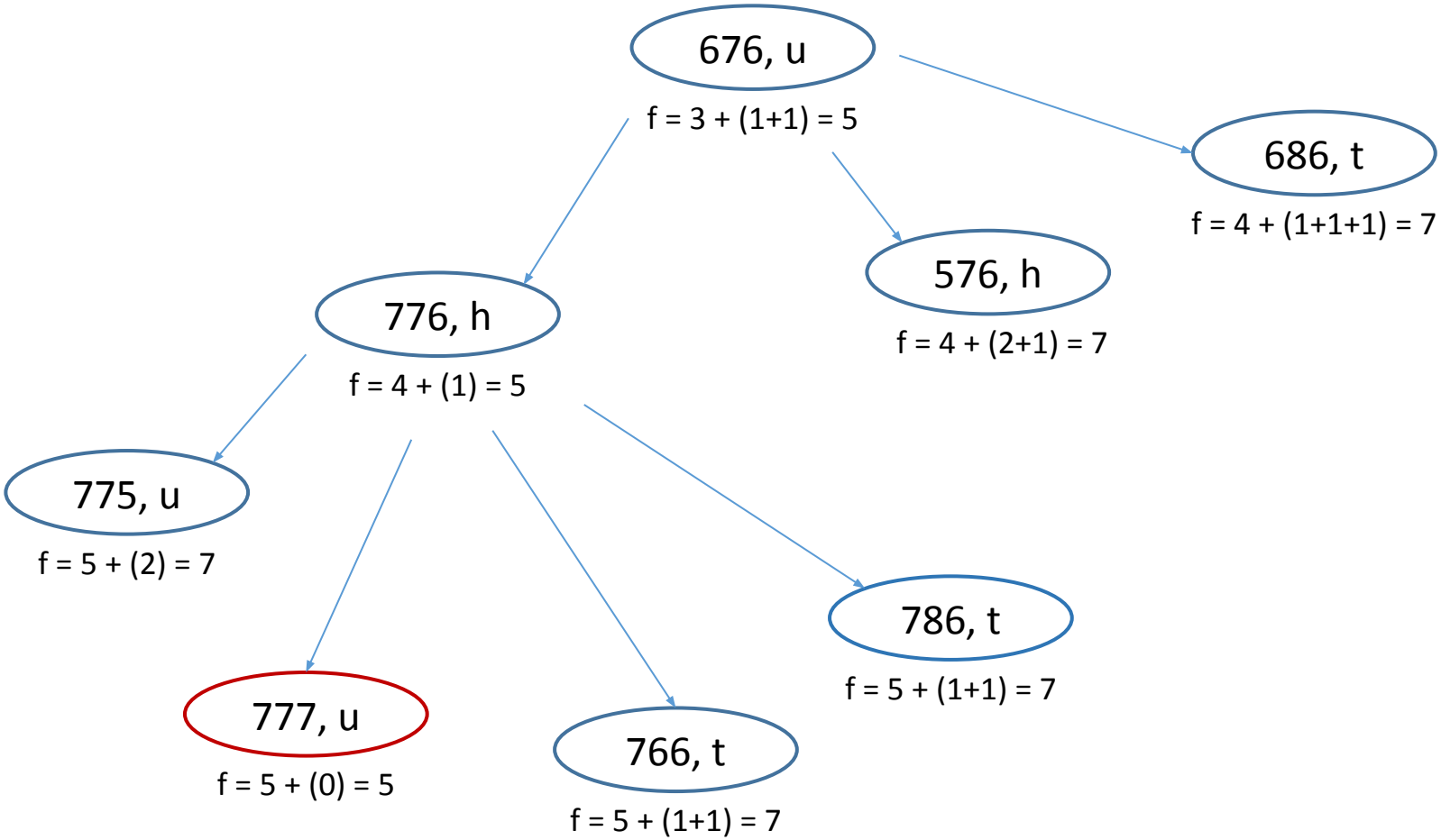
G=777, Forbidden set: {666, 667}



Exercise: Forbidden numbers

A* tree

G=777, Forbidden set: {666, 667}



Exercise: Numbers ordering

The following numbers are to be put in ascending order

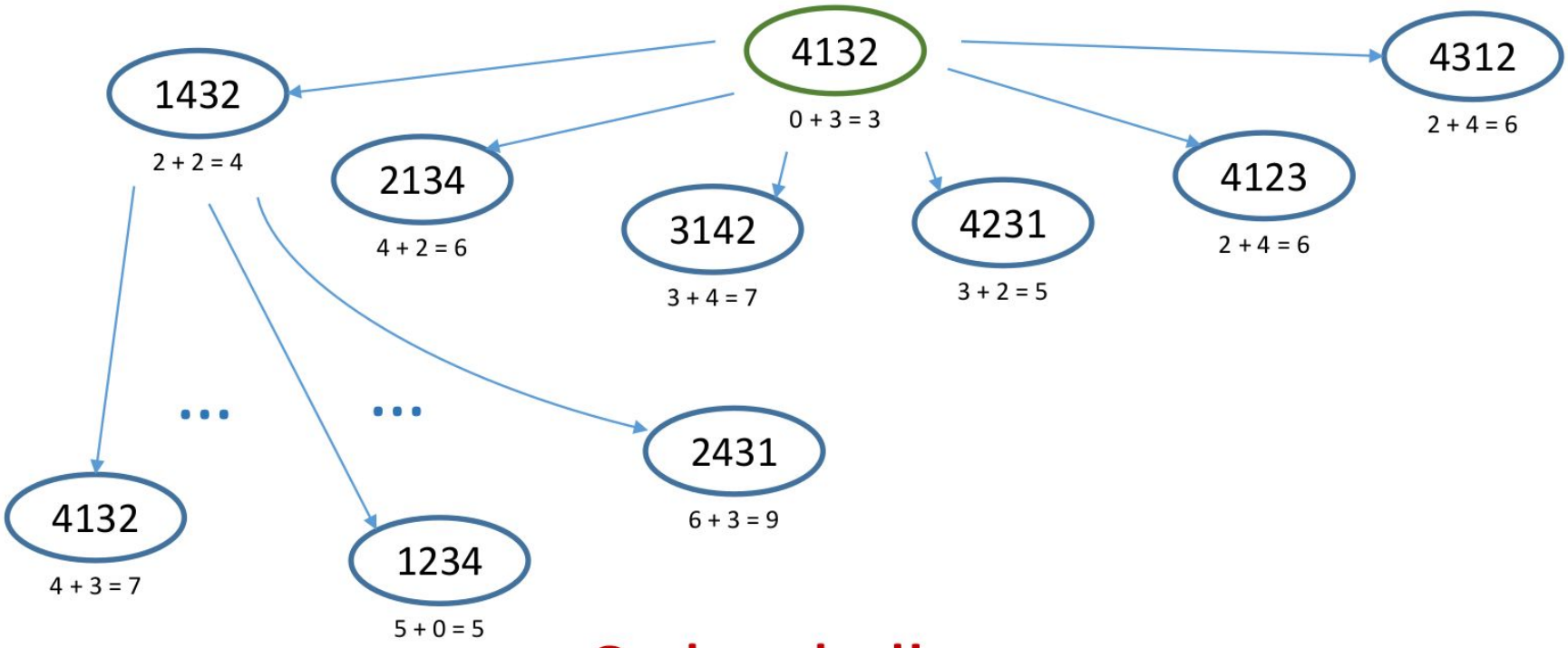
$$4, 1, 3, 2$$

- At each step, while performing the reordering, it is possible to exchange the number in the i-*th* position, with the number in the j-*th* position
- Assume the **cost** of each move is $|j - i| + 1$
- Consider as the heuristic function **h(n)** the number of **misplaced numbers** with respect to the final position

Exercise: Numbers ordering

Goal state: 1,2,3,4

g(s) = |i - j| +1, h(s) = number of misplaced numbers



Goal reached!