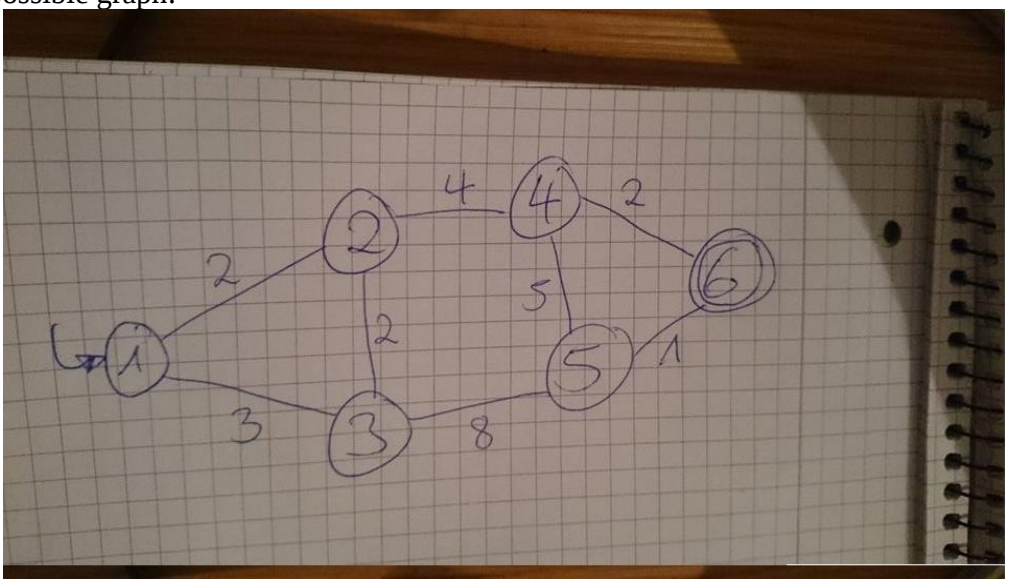
Grundlagen der Wissensverarbeitung

Blatt 2

Daniel Speck, Lena Niermeyer

23.10.2015

1.1.1
Example: Bus Network.
We assume an undirected (on streets it is possible to drive in both directions) graph. [Also possible: Train network with directed edges.]
The nodes are the bus stops. The edges are the streets. The edges are weighted with the minutes it needs to take the distance between bus stops.
Start node is the start point (bus stop) from a bus guest and end node is the destination bus stop.
We are searching for the shortest way (measured in minutes) to get from start to destination.
Picture of possible graph:



The state space is a set of all possible configurations (all the possibilities to cross the edges), but we want to cross every node only one time and take the shortest way. This is possible with Dijkstra's algorithm (no further explanation, this was topic in mathematics 1 and optimization, so we use this knowledge as presupposed knowledge).

Exercise 1.1: (Search Space 1)

2.) a)

Model:

x ≈ Liters in 3-Liter jug

y ≈ Liters in 4-Liter jug

(x,y) ≈ x Liters in 3-Liter jug and y Liters in 4-Liter jug

Transitions:

p ≈ pump, d ≈ drain, III ≈ 3-liter jug, IIII ≈ 4-liter jug

p → III ≈ source is pump and destination is 3-liter jug

Possible states	P → III, conditions: x < 3 | loss or win of liters		P → IIII, conditions: y < 4 | loss or win of liters		III → IIII, conditions: x > 0, y < 4 | loss or win of liters		IIII → III, conditions: y > 0, x < 3 | loss or win of liters		III → d, conditions: x > 0 | loss or win of liters		IIII → d, conditions: y > 0 | loss or win of liters	
00	30	+3	04	+4	/	/	/	/	/	/	/	/
01	31	+3	04	+3	/	/	10	0	/	/	00	-1
02	32	+3	04	+2	/	/	20	0	/	/	00	-2
03	33	+3	04	+1	/	/	30	0	/	/	00	-3
04	34	+3	/	/	/	/	31	0	/	/	00	-4
10	30	+2	14	+4	01	0	/	/	00	-1	/	/
11	31	+2	14	+3	02	0	20	0	01	-1	10	-1
12	32	+2	14	+2	03	0	30	0	02	-1	10	-2
13	33	+2	14	+1	04	0	31	0	03	-1	10	-3
14	34	+2	/	/	/	/	32	0	04	-1	10	-4
20	30	+1	24	+4	02	0	/	/	00	-2	/	/
21	31	+1	24	+3	03	0	30	0	01	-2	20	-1
22	32	+1	24	+2	04	0	31	0	02	-2	20	-2
23	33	+1	24	+1	14	0	32	0	03	-2	20	-3
24	34	+1	/	/	/	/	33	0	04	-2	20	-4
30	/	/	34	+4	03	0	/	/	00	-3	/	/
31	/	/	34	+3	04	0	/	/	01	-3	30	-1
32	/	/	34	+2	14	0	/	/	02	-3	30	-2
33	/	/	34	+1	24	0	/	/	03	-3	30	-3
34	/	/	/	/	/	/	/	/	04	-3	30	-4

Answer:

There are two ways of solving the riddle:

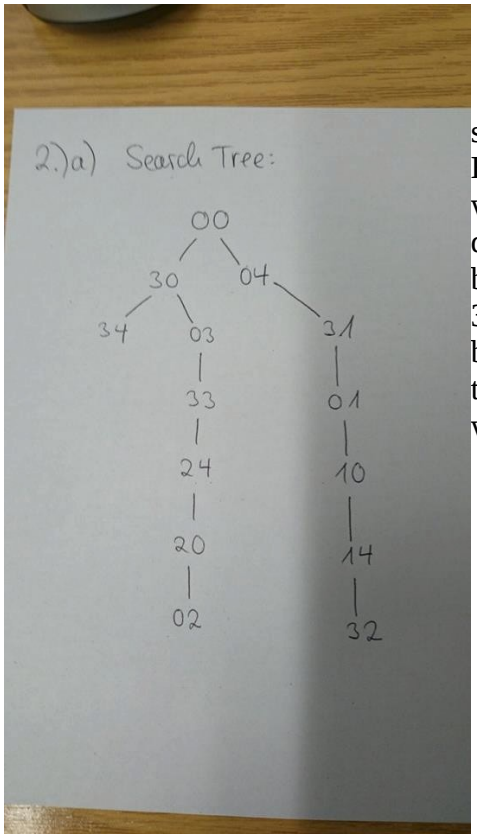
Method 1:

- Fill the 3 liter jug from the pump.
- Empty the 3 liter jug into the 4 liter jug.
- Fill the 3 liter jug from the pump.
- Fill the 4 liter jug with 1 liter from the 3 liter jug - leaving 2 liters in the 3 liter jug.
- Pour away the contents of the 4 liter jug.
- Empty the 2 liters in the 3 liter jug into the 4 liter jug. Leaving 2 liters in the 4 liter jug.
- (Loss of Water: 4 Liter)

Method 2:

- Fill the 4 liter jug from the pump.
- Empty the 3 of 4 liter of the 4 liter jug into the 3 liter jug, leaving 1 liter in the 4 liter jug.
- Pour away the contents of the 3 liter jug.
- Empty the 4 liter jug into the 3 liter jug.
- Fill the 4 liter jug from the pump.
- Empty 2 of 4 liters in the 4 liter jug into the 3 liter jug. Leaving 2 liters in the 4 liter jug.
- (Loss of Water: 3 Liter)

Way to do it:

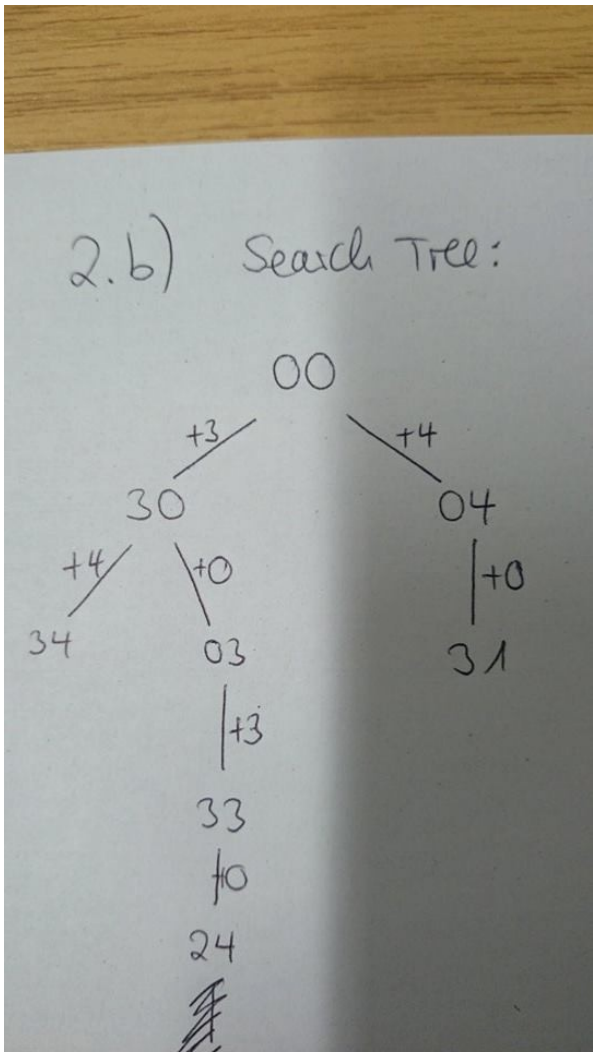


Design a search tree with root (0,0). The table with possible states shows us, what possible children the root has: 30 and 04. Before we enter the new generation, we check the search tree with breadth-first search for duplicates (we don't want duplicates). 30 and 04 have not appeared until now, so they can be children of 00. Next we check 30: possible states after 30 are 34, 03 and 00. 00 did appear in the beginning, is found by breadth-first search and does not appear as child of 30. We do this so long until we reach states of the form (x,2), where y=2, which means our goal “2 liters in the 4 liter jug”.

2.) b)

No, the riddle is unsolvable then.

We do what we did in 2.a), but we eliminate transitions from the last 2 columns, because it's not allowed to drain the wine. Breadth-first search then gives us the search tree below, where the goal (x,2) does not appear.



source: https://www.cs.berkeley.edu/~bh/pdf/v1ch14.pdf

Aufgabe 2.2 (Search Space 2)

In our last assignment one given example for AI scenarios was "navigating through a labyrinth". Mazes/Labyrinths could be found in many computer games and most path finding solutions for AIs in such games could solved via searching.
Pacman, a classic computer game, was one of our examples for computer games with mazes. The player has to navigate Pacman through the maze and collect points represented by little dots. Considering the very basic version of Pacman the play area has about 150 fields, each field has one point by default and the point can be collected by Pacman. Additionally there are 4 ghosts chasing Pacman.
For a solution utilizing searching this game could be modeled by a graph. Each node would represent one of 6 possible booleans:

- boolean1: point is present (not yet collected) or point is already collected by Pacman
- boolean2: Pacman is at this location or not
- boolean3 to 6: Ghost1, ..., Ghost4 is at this location or not

150 locations with 6 booleans each results in $150^6 = 11,390,625,000,000 \approx 1.139 * 10^{13}$ nodes for representing every possible state in the game. The edges would be the transitions between the states, so basically the movements of Pacman and the Ghosts as well as the action of collecting a point. There would be only one start state and therefore only one start node in the graph in the very basic version of Pacman, because in this version Pacman always starts at the same location: some points downwards from the center. The ghosts always start in the center.
The graph has multiple end states, because every node where the "Pacman boolean" is true (so Pacman is present at this state/node) and the at least one of the "Ghost booleans" is true, the game ends by reason of Pacman has been eaten by the Ghosts and therefore the game is lost.
Furthermore every node where all "point is present"-booleans are false (so Pacman has collected all available points) is an end state, because Pacman has won the game.