Homework 01

Submission Notices:

- *Conduct your homework by filling answers into the placeholders given in this file (in Microsoft Word format). Questions are shown in black color, instructions/hints are shown in italic and blue color, and your content should use any color that is different from those.*
- *After completing your homework, prepare the file for submission by exporting the Word file (filled with answers) to a PDF file, whose filename follows the following format,*
  <StudentID-1>_<StudentID-2>_HW01.pdf (Student IDs are sorted in ascending order)
  E.g., ***1852001_1852002_HW01.pdf***
  and then submit the file to Moodle directly WITHOUT any kinds of compression (.zip, .rar, .tar, etc.).
- *Note that you will get zero credit for any careless mistake, including, but not limited to, the following things.*
  1. *Wrong file/filename format, e.g., not a pdf file, use "-" instead of "_" for separators, etc.*
  2. *Disorder format of problems and answers*
  3. *Conducted not in English*
  4. *Cheating, i.e., copy other students' works or let the other student(s) copy your work.*

Problem 1. (2pts) Give a PEAS description for each of the following activities

a. A tailor is sewing clothes on the sewing machine.

Please write your answer in the following table.

Performance measure	the sewing line follows the intended line (usually marked by chalk) exactly, correct clothing size, reduce mistakes.
Environment	sewing thread, needle, sewing machine, cloth
Actuators	hands (move the cloth during sewing), feet (operating the sewing machine)
Sensors	eyes (track the sewing process)

b. A customer is ordering food on GrabFood by using his smartphone.

Please write your answer in the following table.

Performance measure	Order the food successfully in an acceptable time; Food delivered on time
Environment	the customer, the smartphone, the GrabFood application
Actuators	hands (type keyboard), a monitor to display the content of GrabFood application (restaurant pictures, food pictures, ...) during the ordering process
Sensors	touch screen (and maybe voice assistant also) for entering commands

Problem 2. (0.5pt) While appreciating the great contribution of AI to practical life, it is also noteworthy that there are several AI applications designed for shady purposes. Describe an AI application that you think it may have bad effects to our lives and state your opinions.

Briefly describe the "shady" AI application.

Deepfake uses deep learning techniques to create a synthetic media in which a person in an existing image or video is replaced with someone else's likeness. It can manipulate or generate visual and audio content with a high potential to deceive. References: https://en.wikipedia.org/wiki/Deepfake

State your opinion why it is "bad".

Deepfakes have been widely used for

- Producing (celebrity) pornographic videos: This is highly prohibited in the laws of many countries. Someone's figure may be used without his/her consent.
- Fake news, hoaxes, and financial fraud: Deepfake applications fools people into thinking they are receiving instructions from a trusted individual, and thus making erroneous decisions.

Problem 3. (1.5pts) *The wolf, goat, and cabbage problem*. Once upon a time a farmer went to a market and purchased a wolf, a goat, and a cabbage. On his way home, the farmer came to the bank of a river and rented a boat. But crossing the river by boat, the farmer could carry only himself and one of his purchases: the wolf, the goat, or the cabbage. If left unattended together, the wolf would eat the goat, or the goat would eat the cabbage. The farmer's challenge was to carry himself and his purchases to the far bank of the river, leaving each purchase intact.

There are several ways to formulate the given problem as a search problem. However, your answers must be consistent through questions.

Refer to some example formulations:

- https://www.it.uu.se/edu/course/homepage/ai/vt08/Lecture_2/
- https://www.uni-weimar.de/fileadmin/user/fak/medien/professuren/Webis/teaching/ws14/search-algorithms/wolf-goat-cabbage.html
- https://en.wikipedia.org/wiki/Wolf,_goat_and_cabbage_problem

Formulate the above problem as a search problem by answering the following questions.

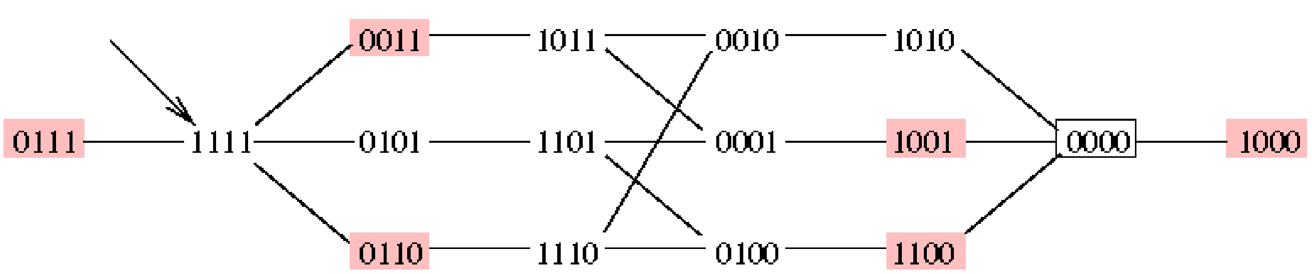
- How do you represent a state? That is, which elements are included in a single state and what is the range of value for each element? (0.5pt)

Model the state by 4 bits (for boat, cabbage, goat and wolf). A 1 means that the item is on this bank, a 0 means it is on the other bank.
For example, the initial state is 1111 and the goal state is 0000

- How many states are there in state space? Justify your answer. (0.5pt)

4 bits, each of which can be either 0 or 1. Thus, $2^4 = 16$ states
Note that, only 10 of those are legal states. Students do not need to explicit represent this.

- Draw a directed acyclic graph (DAG) of the state space. Mark the optimal solution. (0.5pt)



There are some restrictions:

- Restriction 1 is in the state model: "the farmer" is not modeled separate from the boat.
- Restriction 2 determines the possible moves. All moves can be made in both directions.
- Restriction 3 is in the shaded states: these are not allowed.

Students need to mark one of the following two optimal solutions:

- 1111 → 0101 → 1101 → 0001 → 1011 → 0010 → 1010 → 0000
- 1111 → 0101 → 1101 → 0100 → 1110 → 0010 → 1010 → 0000

Problem 4. (1pt) Consider the following 8-puzzle initial state (a) and goal state (b).

2	8	3
1	6	4
7		5

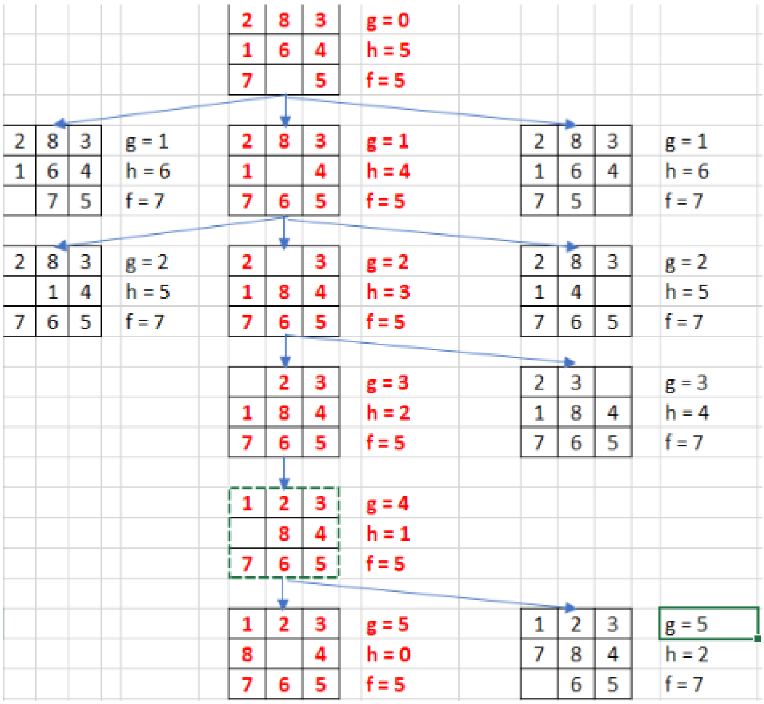
(a)

1	2	3
8		4
7	6	5

(b)

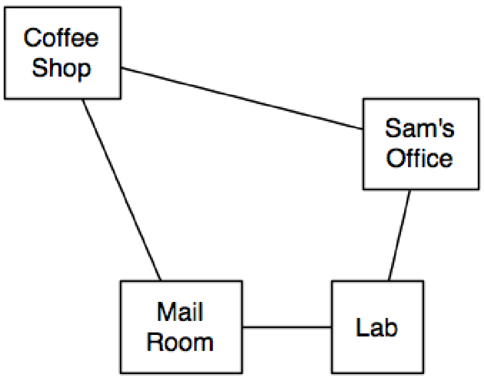
Apply A* using Manhattan distance heuristic function.

- Draw the search tree including possible expanded states during the algorithm procedure.
- Compute the triple (g, h, f) for each state. Mark the optimal strategy found.



Problem 5. (2.5pts) Consider a delivery robot world with mail and coffee to deliver.

Assume a simplified domain with *four locations* as shown aside. This domain is quite simple, yet it is rich enough to demonstrate many of the problems in representing actions and in planning.



The robot, called Rob, can *pick up coffee at the coffee shop, pick up mail in the mail room, move, and deliver coffee and/or mail*. Delivering the coffee to Sam's office will stop *Sam from wanting coffee*. There can be *mail waiting at the mail room* to be delivered to Sam's office.

Rob can *move clockwise (mc)* or *move counterclockwise (mcc)*. Rob can *pick up coffee (puc)* if Rob is at the coffee shop and it is not already holding coffee. Rob can *deliver coffee (dc)* if Rob is carrying coffee and is at Sam's office. Rob can *pick up mail (pum)* if Rob is at the mail room and there is mail waiting there. Rob can *deliver mail (dm)* if Rob is carrying mail and at Sam's office. Assume that it is only possible for Rob to do one action at a time.

Formulate the task above as a search problem by determining the primary concepts.

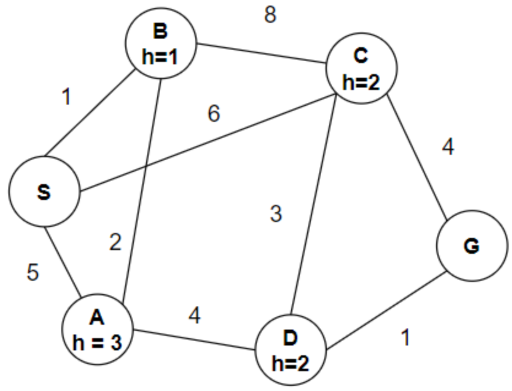
Please write your answer in the table

<table>
<tr><th>Search concepts</th><th>Descriptions</th></tr>
<tr><td>(0.5pt) Representation for a state</td><td>The states are the quintuples specifying the robot's location, whether the robot has coffee, whether Sam wants coffee, whether mail is waiting, and whether the robot is carrying the mail.
For example, the tuple ⟨lab, ¬rhc, swc, ¬mw, rhm⟩ represents the state where Rob is at the Lab, does not have coffee, Sam wants coffee, there is no mail waiting, and Rob has mail.
Another example, the tuple ⟨lab, rhc, swc, mw, ¬rhm⟩ represents the state where Rob is at the Lab, carrying coffee, Sam wants coffee, there is mail waiting, and Rob is not holding any mail.</td></tr>
<tr><td>(0.5pt) State-space graph: how many states there are and how they connect together</td><td>There are 4×2×2×2×2= 64 states. Intuitively, all of them are possible, even if you would not expect that some of them would be reached by an intelligent robot.

Students are not required to draw the graph, however, they must provide examples or general description that can show the characteristics of the state-space graph.</td></tr>
<tr><td>(0.5pt) Set of actions</td><td>There are six actions, ⟨mc, mcc, puc, dc, pum, dm⟩. Not all of which are applicable in each state.</td></tr>
<tr><td>(0.5pt) Transition model</td><td>The complete problem representation includes the transitions for the 64 states.
This following table shows the transitions for two of the states.
Students just need to provide examples or general description that can show the characteristics of the transition model.
<table>
<tr><td>State</td><td>Action</td><td>Resulting State</td></tr>
<tr><td>⟨lab, ¬rhc, swc, ¬mw, rhm⟩</td><td>mc</td><td>⟨mr, ¬rhc, swc, ¬mw, rhm⟩</td></tr>
<tr><td>⟨lab, ¬rhc, swc, ¬mw, rhm⟩</td><td>mcc</td><td>⟨off, ¬rhc, swc, ¬mw, rhm⟩</td></tr>
<tr><td>⟨off, ¬rhc, swc, ¬mw, rhm⟩</td><td>dm</td><td>⟨off, ¬rhc, swc, ¬mw, ¬rhm⟩</td></tr>
<tr><td>⟨off, ¬rhc, swc, ¬mw, rhm⟩</td><td>mcc</td><td>⟨cs, ¬rhc, swc, ¬mw, rhm⟩</td></tr>
<tr><td>⟨off, ¬rhc, swc, ¬mw, rhm⟩</td><td>mc</td><td>⟨lab, ¬rhc, swc, ¬mw, rhm⟩</td></tr>
<tr><td>...</td><td>...</td><td>...</td></tr>
</table></td></tr>
<tr><td>(0.5pt) Path cost</td><td>Since the problem description does not mention the cost for each move, we can simply assume that each action costs 1.</td></tr>
</table>

Problem 6. (2.5pts) Consider the following graph. The start and goal states are **S** and **G**, respectively.

For each of the following **graph search** strategies, work out order in which states are expanded, as well as the path returned. In all cases, assume ties resolve in such a way that *states with earlier alphabetical order are expanded first*. Remember that in graph search, a state is expanded once.



a. Depth-first search (0.5pt)
b. Breadth-first search (0.5pt)
c. Uniform cost search (0.5pt)
d. Greedy best first search with the heuristic *h* shown on the graph (0.5pt)
e. A* search with the same heuristic (0.5pt)

Note that:

- Tree-search DFS avoids repeated states by checking new states against those on the path from the root to the current node.
- For DFS, BFS, and GBFS, the goal test is applied to each node when it is generated rather than when it is selected for expansion.

Please write your answer in the table

Score	Algorithms	States Expanded	Path Returned
0.5pt	Depth-first search	SABC(G)	SABCG
0.5pt	Breadth-first search	SABC(G)	SCG
0.5pt	Uniform cost search	SBACD**G**	SBADG
0.5pt	Greedy best first search with the heuristic *h* shown on the graph	SBC(G)	SCG
0.5pt	A* search with the same heuristic	SBACD**G**	SBADG

Nodes in parentheses are optional. Nodes in bold must be present.