Assignment 2 - CISC 352 Artificial Intelligence

This assignment is intended to gauge your understanding of planning and the PDDL modelling system. Your goal will be to implement a domain in PDDL as well as several problems that fall in that domain. You will be given a zip folder, pddl_template.zip, containing a template file for the domain as well as template files for the problems you are expected to implement.

Implement the following Domain and Problem set in PDDL.

Maximum group size for this project is 3. Minimum group size is 1.

Domain Description – The Treasure Hunter

A treasure hunter just walked into a dungeon with a priceless treasure waiting to be found! Looking around, they gather the following easily apparent facts about the dungeon:

- The dungeon contains rooms that are **connected** to corridors allowing the hero to move around.
- The hero moves through the corridors but is only ever **at** a single room.
- Some of the corridors are **risky**.
- There are **locked** corridors with coloured locks.
- Each room can be empty or have any number of keys **in** it. It may also be **messy**.
- The hero can **hold** a single key.
- There are **two-use** keys and **one-use** keys that are limited in use, and **multi-use** keys can be used infinitely many times.
- One of the rooms is the goal with (containing treasure!).

The hero is lucky to have so much information about this dungeon! But they are not that lucky. The corridors can collapse, the keys are scattered, and it's not so easy to figure out a path through the dungeon. Nonetheless, there's treasure at the end!! The hero can perform any of the following actions:

1. The hero can **move** to an adjacent room (connected to a common corridor as the current room) as long as the corridor is not locked and is not collapsed. The corridors with a red lock are risky and will collapse once passed (i.e., can only be passed once). If they move into a room through a risky corridor, then the destination room becomes "messy" with rubble.
2. When holding the correctly coloured key, the hero can **unlock** a corridor it is adjacent to (i.e., to open a blue lock, the hero must be holding the blue key).
    a) 1-use keys can only be used once.
    b) 2-use keys can only be used twice.
    c) Multi-use keys can be used any number of times.
    d) "Used up" keys do not disappear, but rather remain with the hero.
3. The hero can **pickup** a key in their current room if they aren't already holding a key *and* the room isn't messy.
4. The hero can **drop** a key they are holding, and it will then be in the current room.
5. The hero can **clean** any messy room that they are currently in (making it not messy).

Important: please read and reread the above until you understand it completely. Failing to understand the domain will significantly hamper your ability to succeed on this assignment!

The Problems
The problems you need to implement are defined by the following graphs using the following legend:

ICON	REPRESENTATION	ICON	REPRESENTATION
☆	The starting position		The treasure box
	The Treasure Lock		1-Use Treasure key
	The Red Lock		Multi-Use Red Key
	The Yellow Lock		2-Use Yellow Key
	The Green Lock		1-Use Green Key
	The Purple Lock		1-Use Purple Key
	Room		Corridor

Problem 1

Problem 2

Problem 3

key2 – green

key3 – green

key4 – purple

key5 – purple
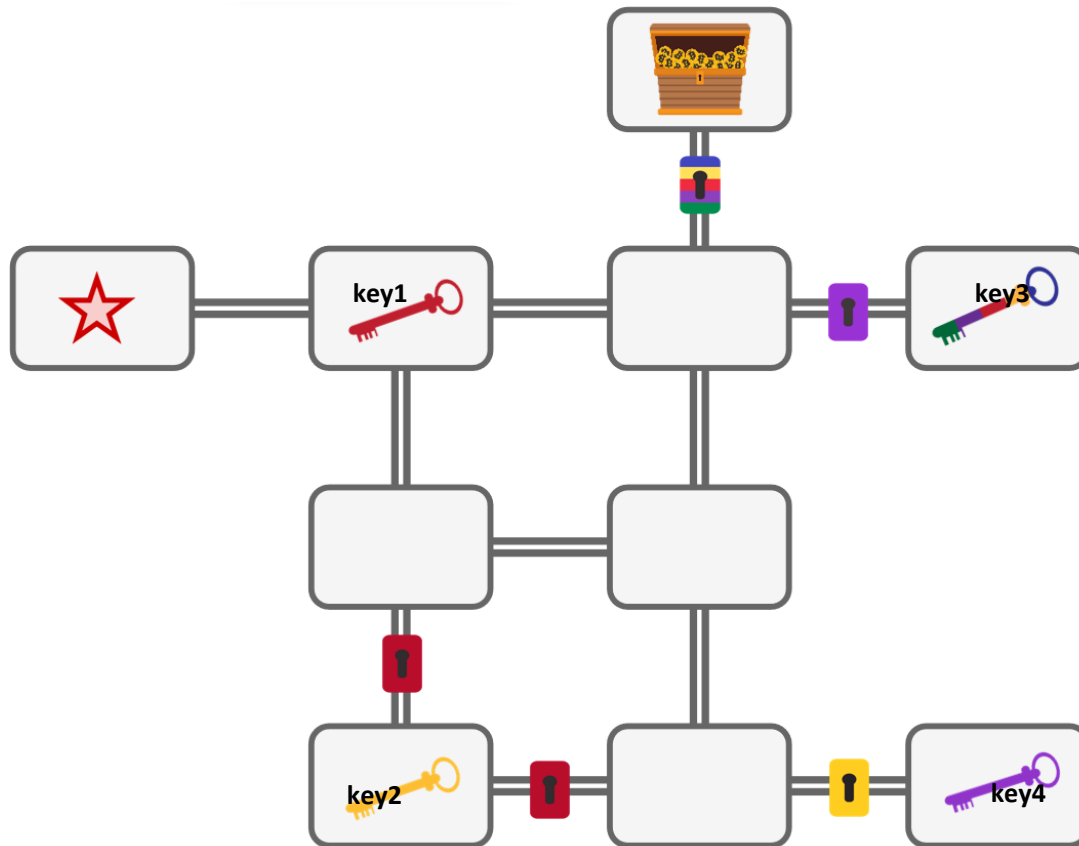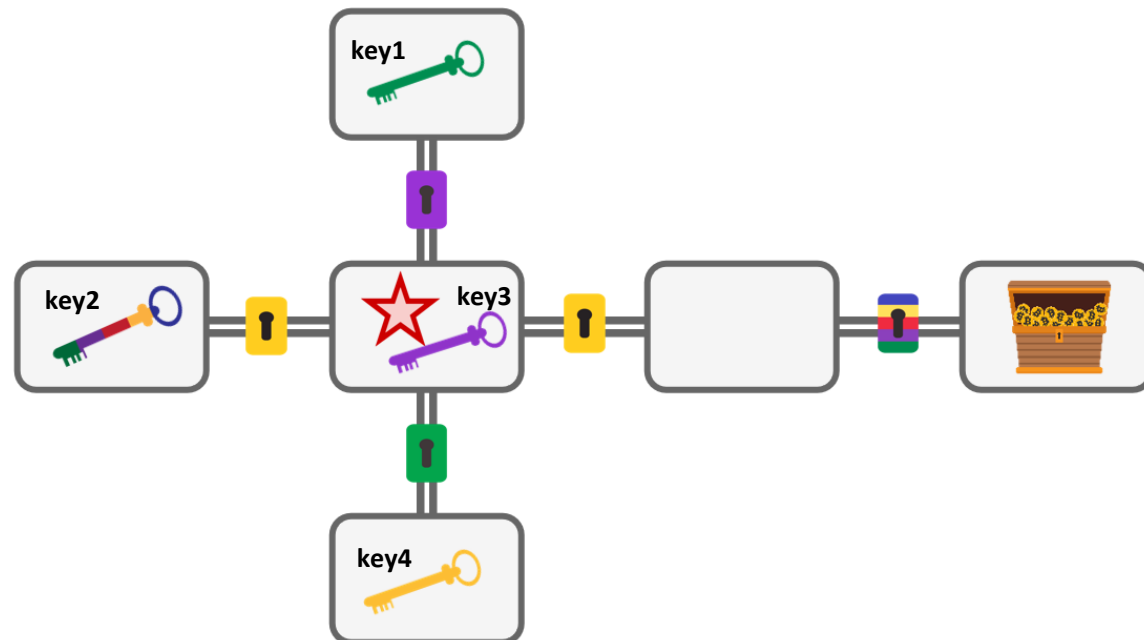
Deliverable & Rubric

PDDL templates will be provided for the domain and problem instances. You will fill these out and submit them for this assignment. Types, constants, objects, and some of the actions will be given. You are responsible for the predicates, action precondi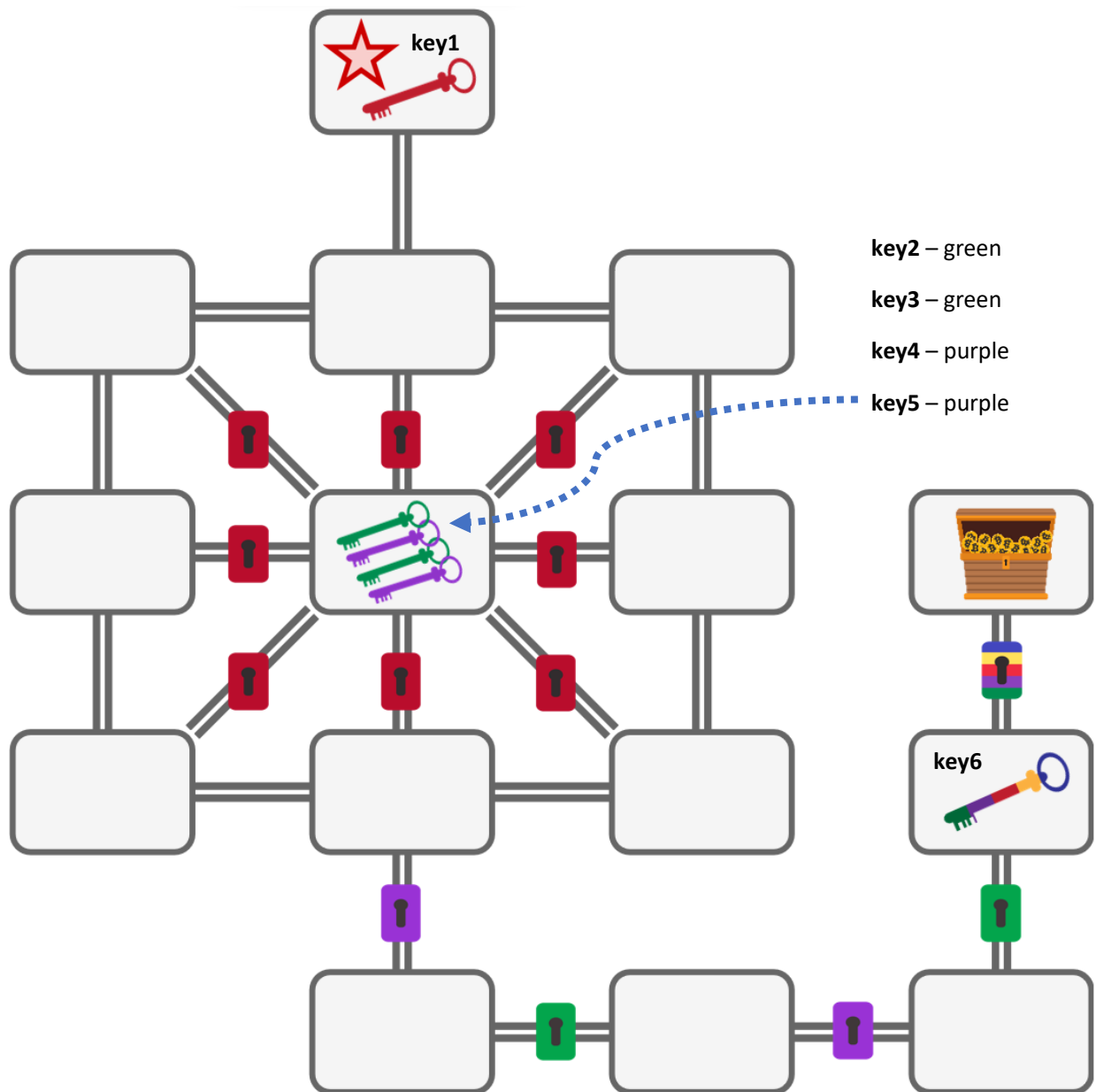tions/effects, initial state, and goal state for the domain and problem files. The first three problems are provided visually above. The final element is to create your own problem file / setting, and ensure that it has the following properties:

1. The number of rooms is between 6 and 9

2. The minimum number of move actions required is greater than 20

3. There is a locked corridor of every colour

In order to hand in this assignment, place your domain file and 4 problem files in a zip folder along with a .docx file, a .txt file, or a .pdf file containing the list of students who worked on the project, their student numbers, NetIDs, and a short description of who contributed what. Upload this .zip file to the assignment page on OnQ.

This assignment is worth **3%** of your final grade in this course. Marks for this assignment will be distributed as follows:

- Correct encoding of the domain (i.e., the actions). **1.75% (0.5%** for **move** and **unlock** actions; **0.25%** for **pick-up, drop,** and **clean)**
- Correct encoding of each problem. **0.75% (0.25% per problem)**
- Newly provided "difficult" problem. **0.25%**
- Readability of submitted PDDL (indentation, comments, etc.). **0.25%**
- Note that while you'll need to create your own fluents, the correctness of these will be assessed as part of the action PDDL you write – there may be several ways to model things.
- The assignment is due on **February 13th, 2025.**

Extra notes to remember:

- You can only use the negative preconditions/effects, and conditional effects (i.e. you **cannot** use *forall*, *exists*, *or*, *imply*, etc. You **can** use the *not* and *and* syntax)
- Use the given corresponding key numbers and colours provided by the figure for each problem.
- Make sure that when naming predicates, they should not use the same name as types.

Useful Resources

To self-check your submissions, here is a tutorial on model alignment: PDDL model alignment

Visual Studio and VSCode are recommended tools for this project, as they have excellent PDDL integration. Alternatively, go to editor.planning.domains in your web browser. Here is a video demonstrating the capability of both:

https://www.loom.com/share/16c132a5092d4bcd81944ddeb9e13176

Note: The video was recorded in a previous year, so the example PDDL is *not* for this assignment.