# Lab 2: Problem Solving by Search

By Sam, Lily Hsu, Lily Chiu, last modified on 2/25, 2019.

This tutorial will introduce some state-space search algorithms, and how they can be used to solve a variety of problems. We start with a simple algorithm and a simple domain: two location vacuum problem. Later we will explore other algorithms and domains.

## Hardware and Software Setup

**Laptop $ ssh [username]@[Workstation’s IP]**

After type password, you will see the workstation logo

**ws $ cd ~/hcc2019/**

**ws $ git clone** [**https://github.com/OpenPPAT/hcc-2019-lab2.git**](https://github.com/OpenPPAT/hcc-2019-lab2.git)

**ws $ cd hcc-2019-lab2**

**ws $ source docker\_run.sh**

**container $ source jupyter\_run\_without\_browser.sh**

[**Lab 2: Problem Solving by Search**](#_74gram21yqax) **1**

[Hardware and Software Setup](#_chc57ow4x0s1) 1

[Overview](#_x5puhc1p2jjo) 1

[Topics and Activities](#_g0fw0bgwxhdt) 2

[Topic/Activity 1 Agent and Environment](#_60rts91qlmd) 2

[Topic/Activity 2 Problem Solving by Search](#_74tpfelx4ci3) 3

## Overview

Estimated Time to Finish: 1 hours

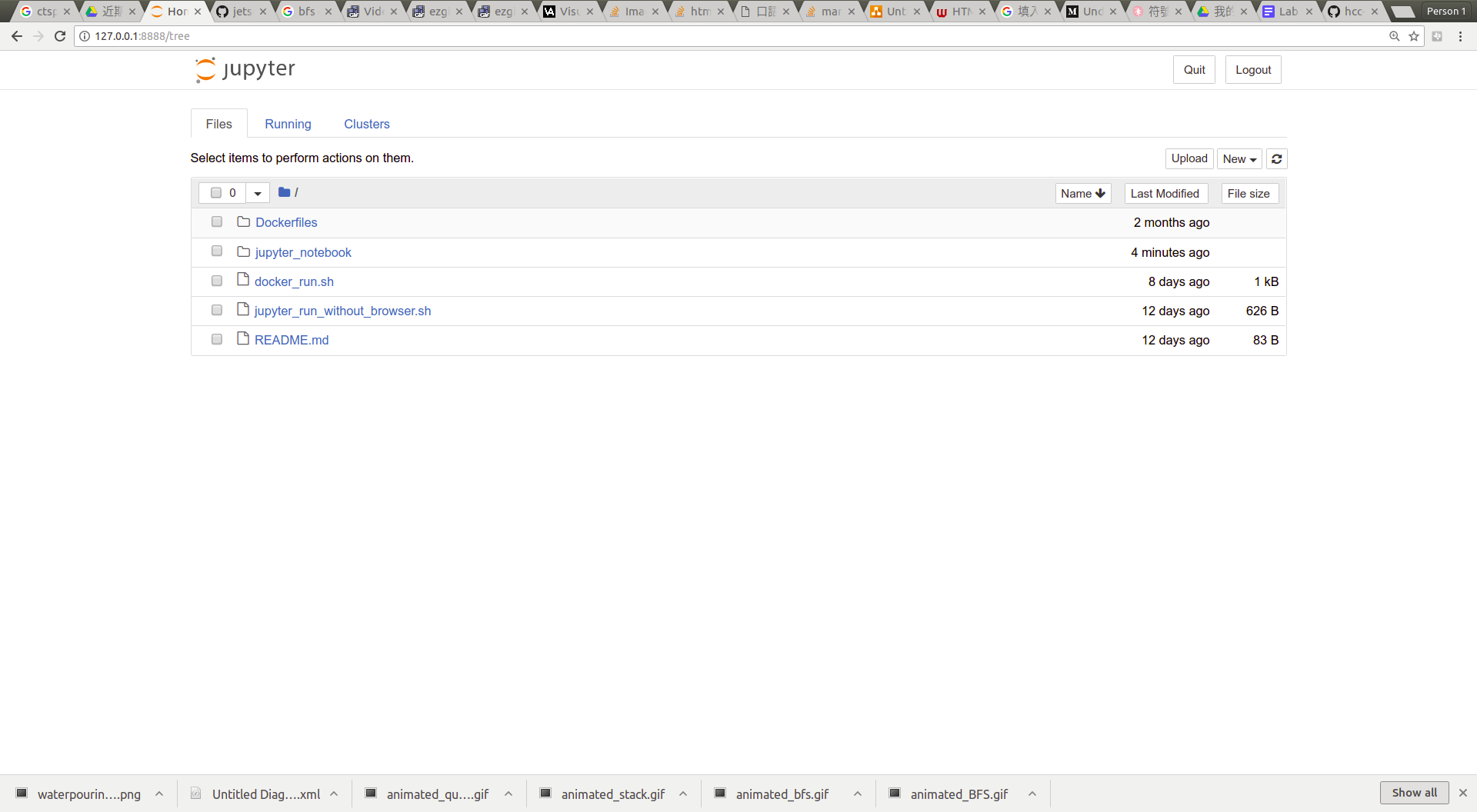
After completing this tutorial you should

* be able to define a problem from high level and solve it by state-space search algorithm.

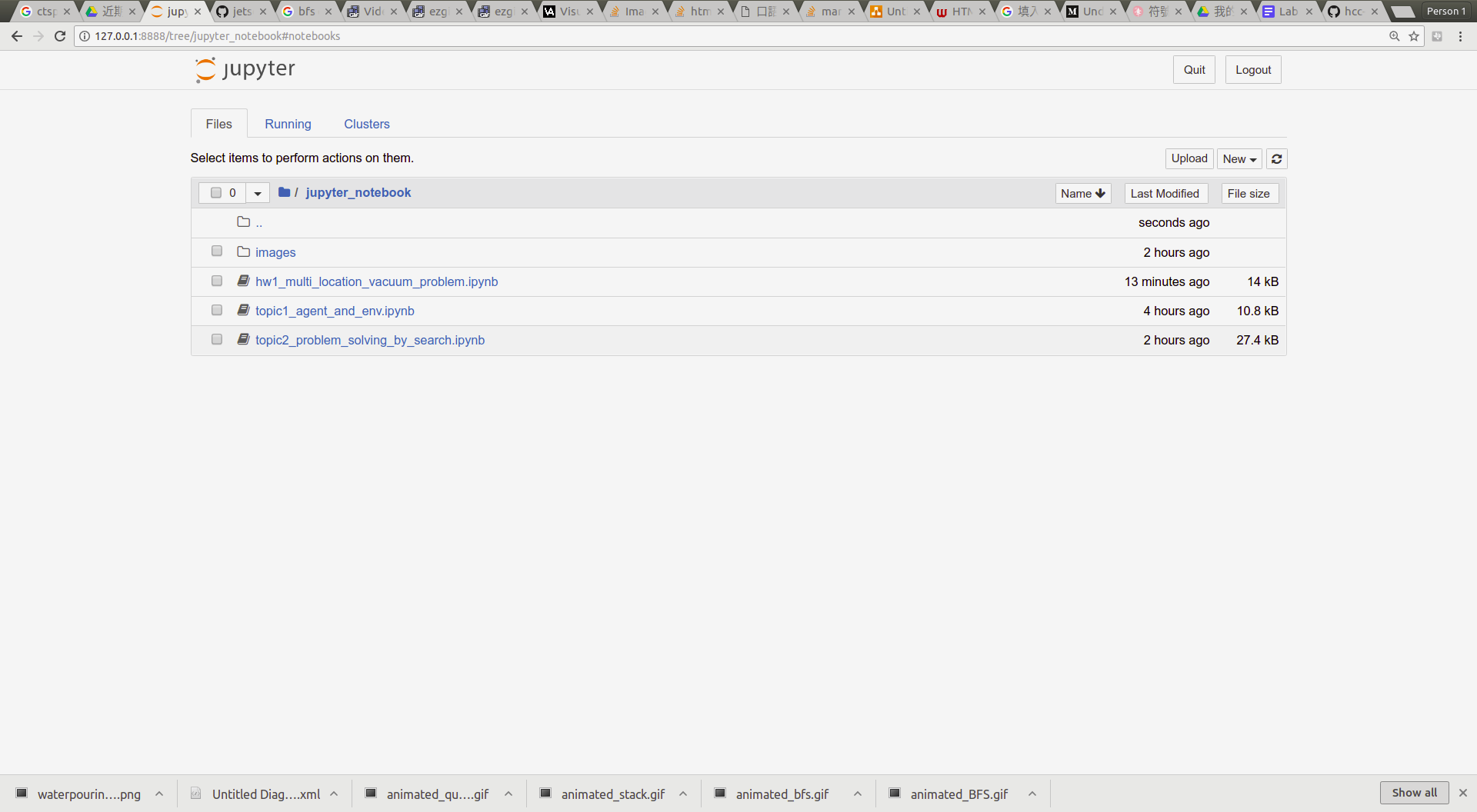
## Topics and Activities

### Topic/Activity 1 Agent and Environment

After launching the jupyter notebook, you will see several files and directory. Our lab tutorials are in the **jupyter\_notebook** directory.



Double click the topic1\_agent\_and\_env.ipynb, and start to run the notebook.



### 

### 

### Topic/Activity 2 Problem Solving by Search

All materials in this section are provide by [jupyter notebook](https://github.com/OpenPPAT/hcc-2019-lab2/blob/master/jupyter_notebook/topic2_problem_solving_by_search.ipynb) and [slide](https://drive.google.com/open?id=1cdRh8OltXpviBy1pRPl_2fLuKmvN-kiWBDzYh-A13R8), please take a look and reply the following questions.

# Discussion:

Discuss the following questions with your teammate. When you finish it, ask TA to check your result.

1. Please explain briefly how to use state-space search to solve the some problem, what's the problem formulation.
2. Please verify the code to solve **RouteProblem** by **uniform cost search**, and show the comparison about the **step costs** caused by breadth cost search and uniform cost search. explain why.
3. As the end of topic2 document, the state-space problem **usually cannot be solved by depth first search** (infinity cost means cannot solve the problem). please explain why?

# 

# Howework:

Please see **hw1\_multi\_location\_vacuum\_problem.ipynb.** Like example 2-1, but we want to add mobility of the vacuum robot to clean the place. By expanding to two dimensional space and up to 9-places, the problem may be more complex. In this question, you need to define the problem which is **limited by some rule**.

Please compress your .ipynb file into .zip file and upload it to New E3 with filename: **hw2\_[student\_id].zip,** The **expect result is shown below**, also find that in [[here](https://github.com/OpenPPAT/hcc-2019-lab2/blob/master/jupyter_notebook/hw1_multi_location_vacuum_problem.ipynb)]

**Deadline is 3/28 23:55**



