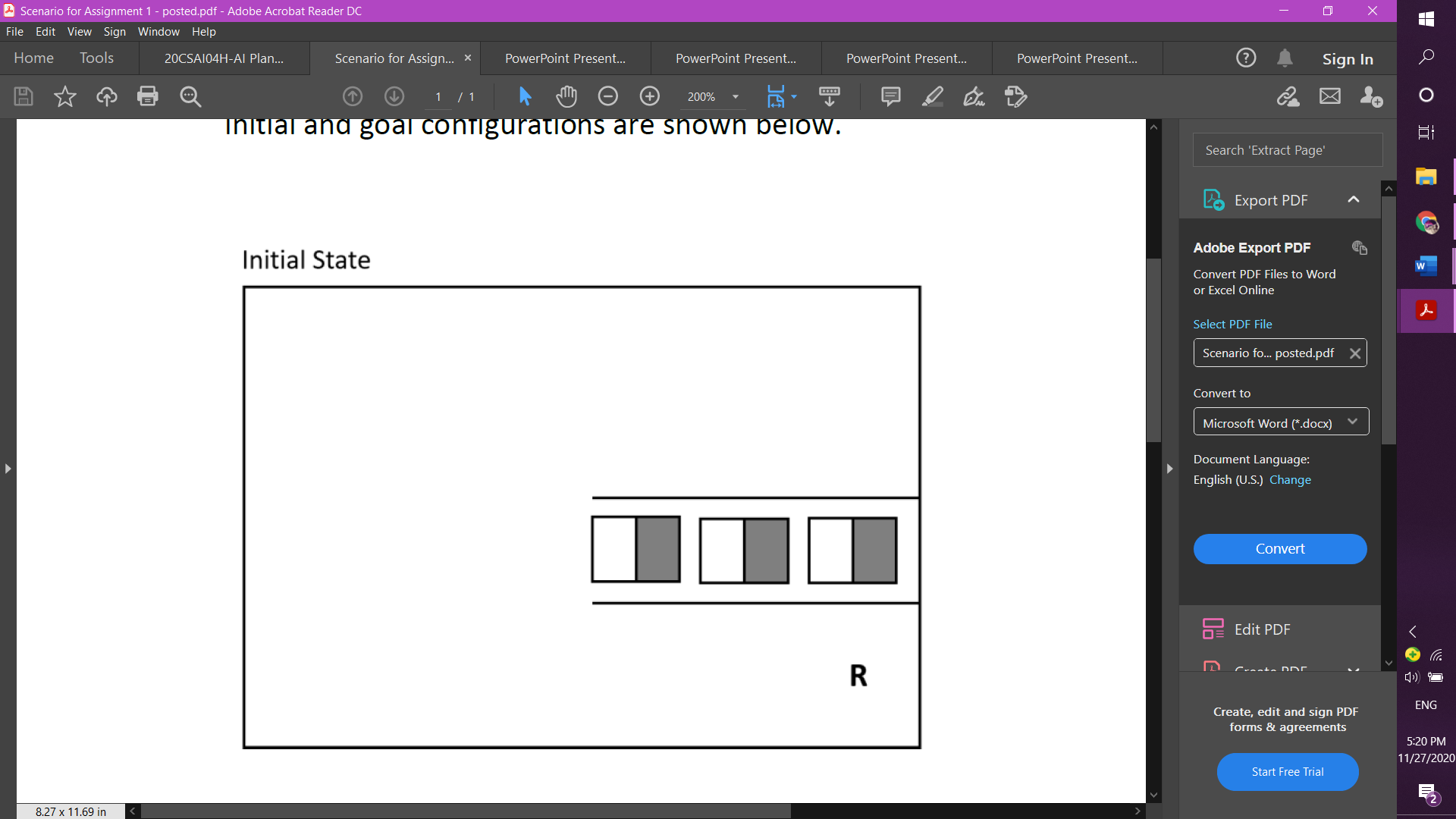
**Assignment 2 AI planning for robot systems**

**The robot world:**



D1way

D2way

D3way

start

B1way

B1 B2 B3

D1

D2

D3

environment = {

'B1': ['B1way', 'B2'],

'B2': ['B1', 'B3'],

'B3': ['B2'],

'D1': ['D1way'],

'D2': ['D2way'],

'D3': ['D3way'],

'D1way': ['D1','B1way','D2way'],

'D2way': ['D2','D1way','D3way'],

'D3way': ['D3','D2way'],

'B1way': ['D1way','B1','start'],

'start': ['B1way']

}

**Roadmap representation as a graph:**

D3 D3way

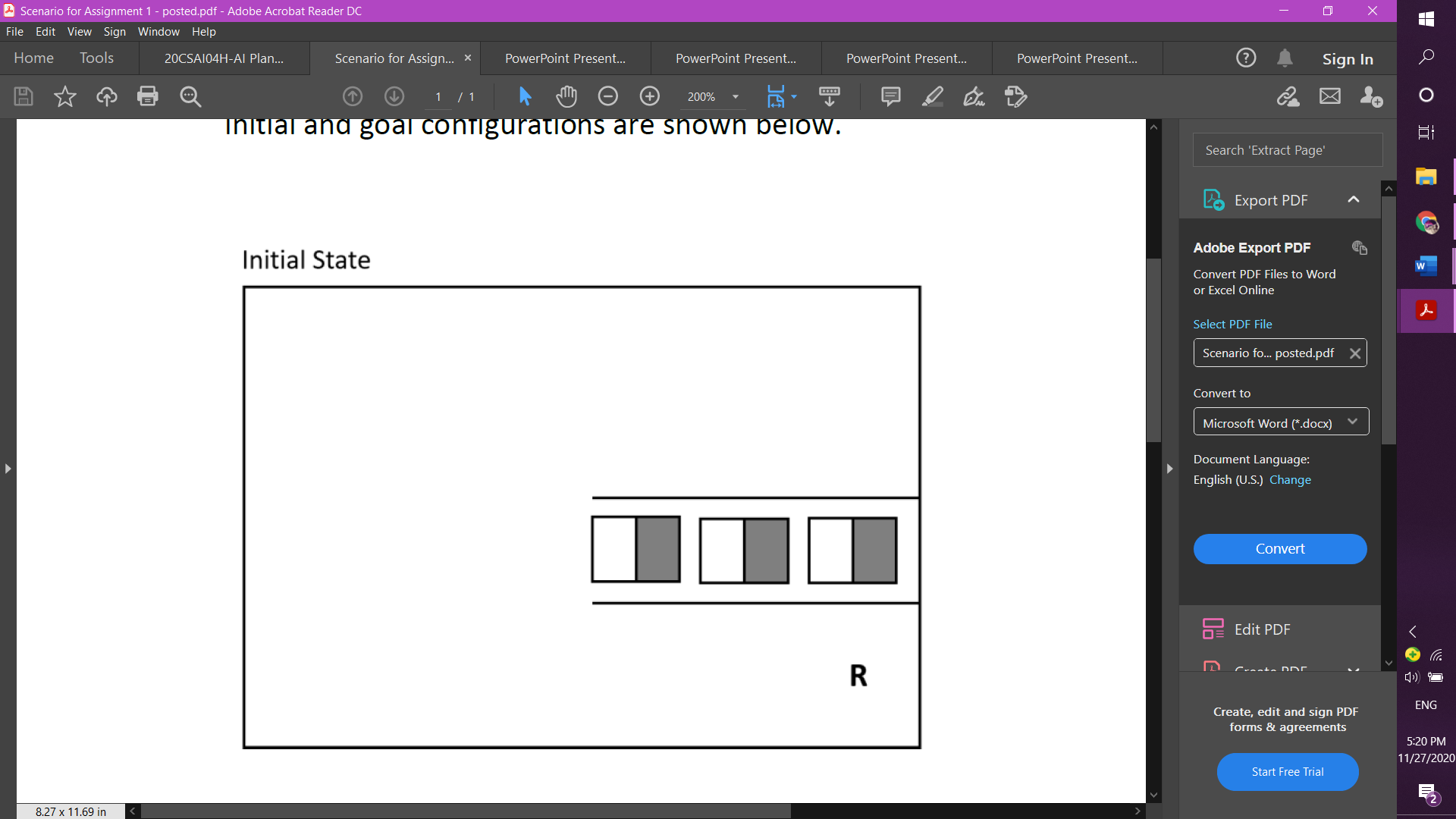
D2 D2way

B1way B1 B2 B3

D1 D1way

Start

**Configuration space:**



C obs

C free

D1way

D2way

D3way

start

B1way

B1 B2 B3

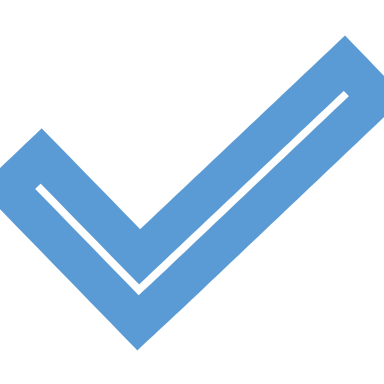
D1

D2

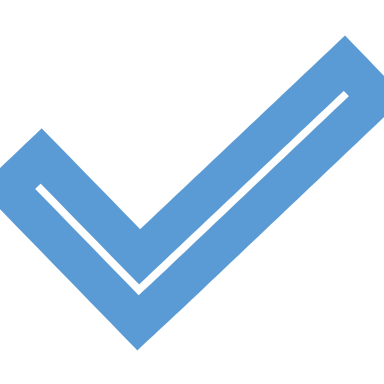
D3

C space = C free and C obs

**Is the graph G called a roadmap?** Let’s check if it satisfies these two conditions:

1. **Accessibility**: it is always possible to connect a q I (start) and q G (goal) to some S 1 and S 2 respectively in S 

2. **Connectivity preserving**: If there exists a path in C free from S 1 to S 2, then

there exists a path in G from S 1 to S 2 

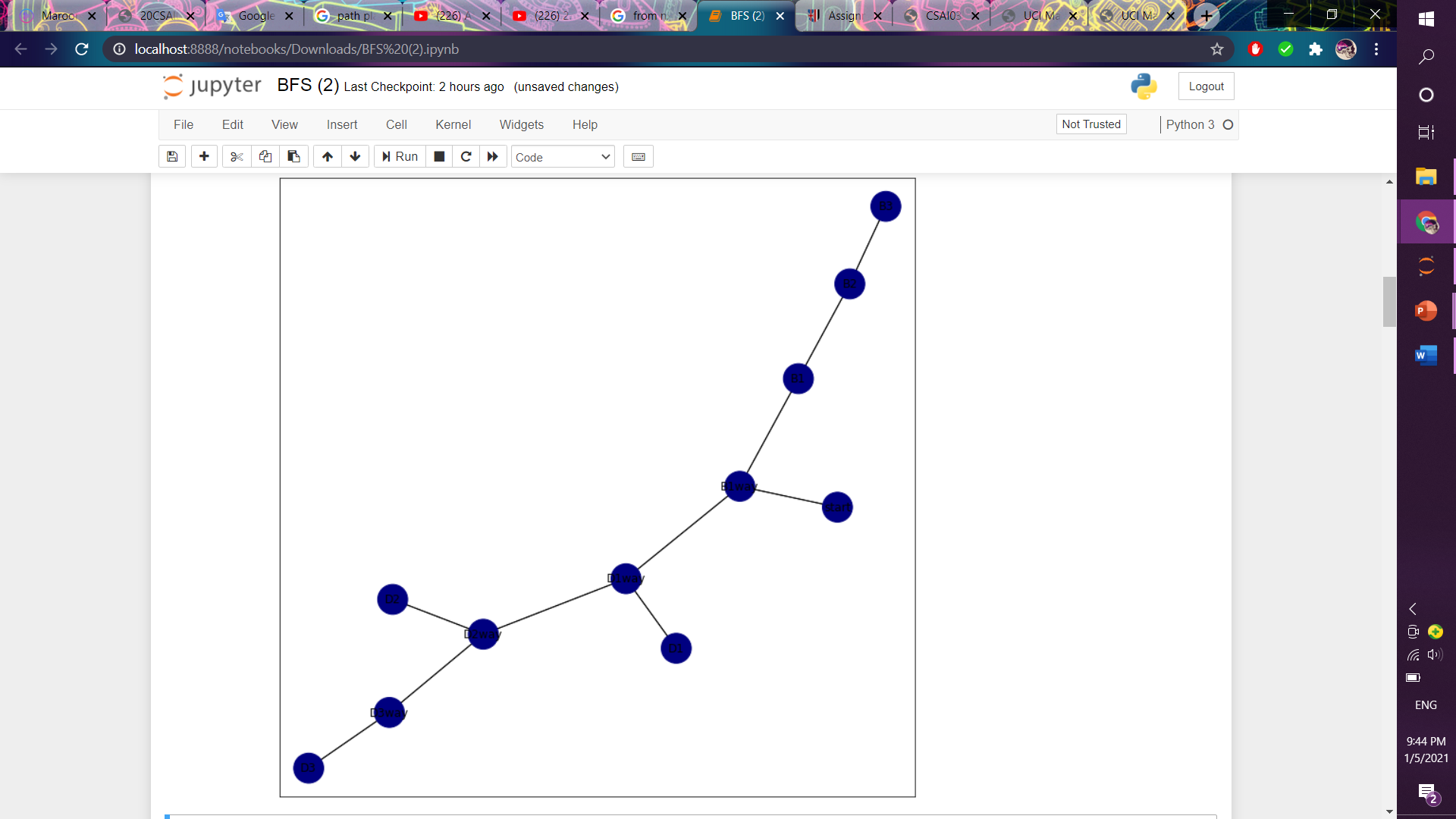
**The data structure representing the world is as a:** dictionary

**Single query sampling-based planning:**

Using the planning steps, I will explain my plan.

**Step 1: Initialization:**

This is undirected graph G contains the environment as nodes and the paths as edges.



**Step 2: Vertex selection method (VSM):**

I used BFS algorithm to find the shortest path from node to another.

**Step 3: Local planning method (LPM):**

When each box gets moved to its destination the nodes of the boxes will no longer be an obstacle in the other hand when the destinations are filled with boxes, they will become obstacles (this is represented in the code by color changing).

**Step 4: Insert an edge in the graph:**

The edges will get added from the environment to the graph.

**Step 5: Check for a solution:**

Check if G got the best path.

**Step 6: Return to step 2:**

Return to step 2 till all the boxes get moved and the goal is achieved.

**Successes:**

The goal of the robot has successfully reached by finding the shortest path from the start to B1 then from B1 to D1 to put down the box 1 in destination 1 then again till all boxes get moved to there destinations using graph representation and BFS algorithm, the movement of the boxes is represented by color changing, the 3 boxes are red in the first iteration then when each box is moving the destination become red instead of blue and the box become blue instead of red until all boxes get moved so all destinations are red. The search\_BFS function can solve any problem even if you changed the the nodes or the whole environment.

**Failures:**

It fails when there is no environment or it’s not well defined, also it can’t work with search\_ BFS function only because it will find the shortest path from point to another while ignoring that the boxes should be considered as obstacles and they should get moved to other locations.

**Scope of the system:**

The system should deliver a graph representation of the robot environment and find the shortest path using BFS algorithm to move the boxes to its destinations then return to the starting point.

**Lessons:**

I have used mainly lecture 8 and 9 in this report.

**References:**

<https://modernrobotics.northwestern.edu/nu-gm-book-resource/10-2-c-space-obstacles/#department>

<https://books.google.com.eg/books?id=-PwLBAAAQBAJ&pg=PT306&lpg=PT306&dq=Unidirectional+(single+tree)+methods&source=bl&ots=0iyC6qyolp&sig=ACfU3U0myC9RlHH3mczLaAbJxOR3EzHRLg&hl=en&sa=X&ved=2ahUKEwjdlvq6p4XuAhXuSxUIHciwBogQ6AEwEXoECBAQAg#v=onepage&q=Unidirectional%20(single%20tree)%20methods&f=false>

**Name:** Rawan Abdellah