

assignment # 2

Assignment Type

- AI Search Techniques
 - The search algorithms can solve this problem easily so you must implement one of the **uninformed** search techniques and one of the **informed** search techniques to solve the problems

I. Problem #1

1. Problem#1 Overview

Given a list of positive integer **Items** whose elements are guaranteed to be in sorted ascending order, and a positive integer **Goal**, and **Output** is a list of three elements [**A,B,C**] taken from items that together add up to **goal**. The **Output** must occur inside the items list in that order (ascending order).

?-threeSum([3,8,9,10,12,14],27,Output).

Output=[8,9,10];

Output=[3,10,14].

?-threeSum([2,4,8,10,12,14],25,Output).

false.

?-threeSum([2,4,8,12,14,16],20,Output).

Output=[2,4,14].

?-threeSum([1,2,3,4,5,6,7,8,9],12,Output).

Output=[1,2,9];

Output=[1,3,8];

Output=[1,4,7];

Output=[1,5,6];

Output=[2,3,7];

Output=[2,4,6];

Output=[3,4,5].

2. Problem#1 Components:

This program consists of the following engines:

- Planner Engine: is responsible for:
 - Taking the query in prolog entered by the user.
 - Apply the planning algorithm using depth first search or greedy algorithm to get the path. This algorithm should be implemented in prolog.
 - You must solve this problem twice (depth first search and greedy algorithm)

3. Problem#1 Grading Criteria

| | |
|---|---|
| 1 | Accept Input from user (not hard coded). Problem State Representation. |
| 2 | Planner Engine (Using depth first search algorithm) Uninformed search. |
| 2 | Planner Engine (Using greedy algorithm) Informed search. |
| 1 | heuristic function (Informed search) |
| 1 | Correct output. |

II. Problem #2

1. Problem#2 Overview

Daisy loves playing games with words. Recently, she has been playing the following Deletive Editing word game with Daniel.

Daisy picks a word, for example, "DETERMINED". On each game turn, Daniel calls out a letter, for example, 'E', and Daisy removes **the first occurrence** of this letter from the word, getting "DTERMINED". On the next turn, Daniel calls out a letter again, for example, 'D', and Daisy removes its first occurrence, getting "TERMINED". They continue with 'I', getting "TERMNED", with 'N', getting "TERMED", and with 'D', getting "TERME". Now, if Daniel calls out the letter 'E', Daisy gets "TRME", but there is no way she can get the word "TERM" if they start playing with the word "DETERMINED".

Daisy is curious if she can get the final word of her choice, starting from the given initial word, by playing this game for zero or more turns. Your task is to help her to figure this out.

```
% deletiveEditing(Initial,End).
```

Each word consists of at least one and at most 30 uppercase English letters; **Initial** is the Daisy's initial word for the game; **End** is the final word that Daisy would like to get at the end of the game.

```
?- deletiveEditing(['D','E','T','E','R','M','I','N','E','D'], ['T','R','M','E']).
```

True.

```
?- deletiveEditing(['D','E','T','E','R','M','I','N','E','D'], ['T','E','R','M']).
```

False.

```
?- deletiveEditing(['D','E','I','N','S','T','I','T','U','T','I','O','N','A','L','I','Z','A','T','I','O','N'], ['D','O','N','A','T','I','O','N']).
```

True.

```
?- deletiveEditing(['C','O','N','T','E','S','T'], ['C','O','D','E']).
```

False.

```
?- deletiveEditing(['S','O','L','U','T','I','O','N'], ['S','O','L','U','T','I','O','N']).
```

True.

2. Problem#2 Components:

This program consists of the following engines:

- Planner Engine: is responsible for:
 - Taking the query in prolog entered by the user.
 - Apply the planning algorithm using informed search algorithm (**greedy** algorithm) to get the path. This algorithm should be implemented in prolog.

3. Problem#2 Grading Criteria

| | |
|-----|---|
| 2 | Accept Input from user (not hard coded). Problem State Representation. |
| 1.5 | Planner Engine (Using greedy algorithm) Informed search. |
| 1.5 | heuristic function |
| 1 | Correct output. |

Important Notes: (Please read these notes carefully to avoid losing grades)

- Please submit one .pl file containing your solution. The file name must follow this structure: ID1_ID2_ID3_ID4_DEPARTMENT_GROUP.
- The number of students in a team must be 3 or 4.
- Cheaters will be given a NEGATIVE grade and no excuses will be accepted.
- The deadline is after 10 days of the assignment announcement.

Good Luck