CENG 242

Hw #6

Spring 2006/2007

(Due: May 28th, 2007 Monday 23:59)

I think you all know Pokémon. Lots of anime series and video games have been created about these creatures having special powers and abilities. The origin of the Pokémon is Japan and the word Pokémon stands for "Pocket Monster" in Japanese. Both in games and movies, players of the games are designated as Pokémon Trainers, and the two general goals for such Trainers are to complete the Pokédex by collecting all of the available Pokémon species found in the fictional region where that game takes place; and to train a team of powerful Pokémon from those they have caught to compete against teams owned by other Trainers, and eventually become the strongest Trainer, the Pokémon Master. Here is a sample Pokémon called Charmander:

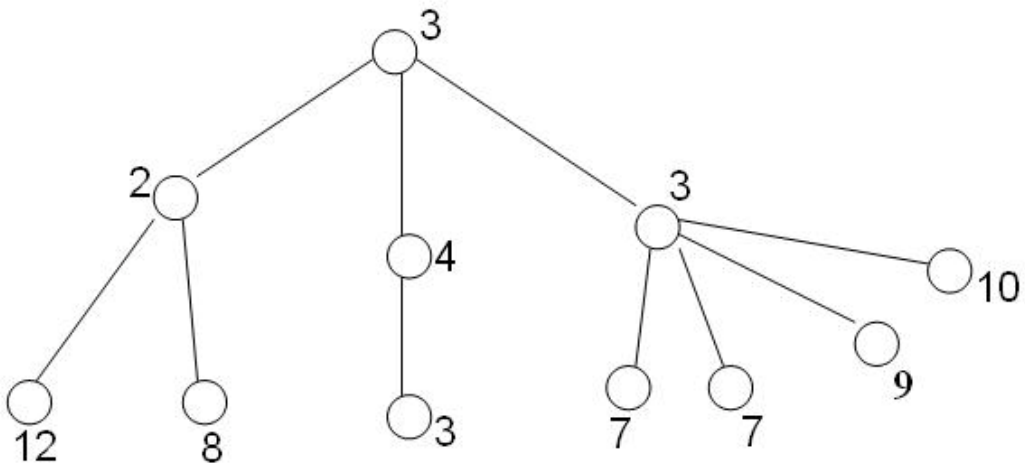
Now, you are a Pokémon Trainer and you need to collect some Pokémons. Luckily you have found a map showing the roads between cities and how many Pokémon species live in each city. To compete in Pokémon competition you need to collect exactly T Pokémon species where T is an integer that will be given to you before competition date.

Rules:

- The map has a tree liked shape.
- The roads between cities are undirected (you can go both directions).
- If you enter a city, you should collect all the Pokémons in the city or you lose.
- You can not enter a city more than once.
- You can start and finish anywhere.
- You should find all possible paths.

Example:

This is a sample map:



If you are asked to collect 15 Pokémon species, all possible paths are:

2 3 3 7
2 3 3 7
3 3 9
7 3 3 2
7 3 3 2
9 3 3

Specifications:

- You should write a Prolog code solving this problem. The predicate you should implement is:

collectPokemon(*T* , *TheMap*, *PossiblePaths*).

For the above example, sample call will be:

?- collectPokemon(15, [3, [2, [12], [8]], [4, [3]], [3, [7], [7], [9], [10]]], PossiblePaths).
PossiblePaths = [2,3,3,7] ? ;
PossiblePaths = [2,3,3,7] ? ;
PossiblePaths = [3,3,9] ? ;
PossiblePaths = [7,3,3,2] ? ;
PossiblePaths = [7,3,3,2] ? ;
PossiblePaths = [9,3,3] ? ;
no

I think it is clear how to represent a tree and how to represent the solution.

- You can give the paths in any order you want, but you should give all of them.
- You will submit a single .pl file **hw6.pl** including all your definitions.
- You will submit your codes through cow system. Specifications (file name, predicate name etc.) are strict. Breaking any of them will cost you getting a 0 since black box method is used.