Middle East Technical University | Department of Computer Engineering

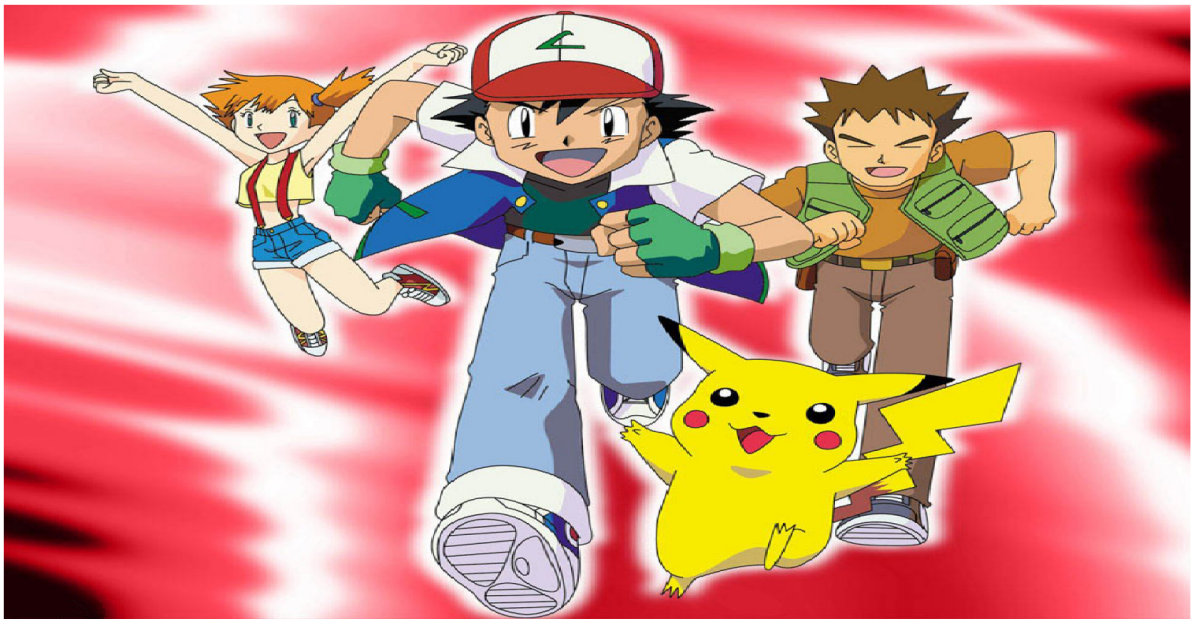
CENG 242

Programming Language Concepts

Spring '2013-2014

Hw2

Due date: 30 March 2014, Sunday, 23:55



1 Introduction

In this homework you will be given a simple social network of pokemon trainers and you need to develop some functions to solve problems related to this social network using Haskell programming language.

2 Social Network

The social network will be a tuple and in the social network you will have two types of data.

The first one is related to trainer information and the other is related to friendship relationships. Trainer information will consist of the followings:

- **Trainer Id :** Each trainer will have a unique id which is an integer.
- **Trainer Name :** Each trainer will have a name information. Trainer name is given as String.

- **Gender:** Each trainer will have gender information, either "female " or "male" which has type String.
- **Age :** Each trainer will have age information as Integer. Age can be at least 0 and at most 100. The inputs will be given accordingly.
- **Pokemons :** This data will be a list of Strings. Each trainer can have more than one pokemon and each of them are given as String.

Friendship relationships will be given as two trainer-id pair. For example a pair like (tid1,tid2) means that trainers having tid1 and tid2 as id's are friends. The friendship relationship is reflexive meaning that if a trainer is friend of the other trainer then the reverse is also true. Friendship relationships will always be given sorted. By sorted we mean that for each pair the first number will be smaller and the second will be larger. Also if the first element of one pair is smaller than the other pairs first element then that pair should come earlier. The last rule about sorting the pairs is when their first elements are same, then we sort according to second elements of the pairs. In the inputs relationships will always be given to you sorted in the described manner. You should also return sorted versions (in increasing order) of the friendship relationships after all of your functions. We will provide you a function to do this. You can use that function to sort the relationships.

An example Social Network can be following, the inputs you will face will be familiar to this (try to remember that the friendships are sorted in pairs and as a list):

([(1 , "Ash" , "male" , 10 ,["bulbasaur" , "charmender" , "pikachu" , "squirtle"]),
(2 , "Misty" , "female" , 11 ,["staryu"]),
(3 , "Gary" , "male" , 10 ,["doduo", "nidoking"]),
(4 , "Brock" , "male" , 18 ,["onix"]),
(7 , "Ritchie" , "male" , 10 ,["pikachu"]),
(9 , "Tracey" , "male" , 14 ,["venonat"])],
[(1 , 2),(1 , 4),(2 , 4),(3 , 9),(4 , 7),(7 , 9)])

3 Functions to be implemented

You will implement a total of 4 functions. The definitions and the information about these functions are provided below:

- `get_fof_network social_network trainerid_pairs_list n`

  You should implement a function named `get_fof_network`. This function takes three parameters social-network (as clarified above), trainerid-pairs-list and n and returns a new modified `social_network`. trainerid-pairs-list is a list containing trainerid pairs. For each pair of pokemon trainer you should find distance between them and if it is smaller or equal to n then you should make these two pokemon trainers friends in the returning social network. If they are already friends you shouldn't add their friendship relationship again. For example if you are given the above social network as input and you got [(1,3)] as `trainerid_pairs_list` and n=3 then you should return the same social network. But in the case of n=4 then you should make 1 and 3 friends in the returning social network. Because the minimum distance between Ash and Gary is 4.

  For the n=4 case your output should be:

  ([(1 , "Ash" , "male" , 10 ,["bulbasaur" , "charmender" , "pikachu" , "squirtle"]),

  (2 , "Misty" , "female" , 11 ,["staryu"]),

  (3 , "Gary" , "male" , 10 ,["doduo", "nidoking"]),

  (4 , "Brock" , "male" , 18 ,["onix"]),

(7 , "Ritchie" , "male" , 10 , ["pikachu"]),

(9 , "Tracey" , "male" , 14 , ["venonat"])],

[(1 , 2), (1 , 3), (1 , 4), (2 , 4), (3 , 9), (4 , 7), (7 , 9)])

Note: In the case of there are more than one trainerid pairs in the second parameter then you should look at the distances of the original social network not the one you just modified.

- `recomendation social_network trainerid distance pokemon` : You should implement a function named `recommendation`. In this function you will return a recommendation list for the given pokemon trainer. You will recommend new friends for the pokemon trainer to fulfill his ambitions and be a pokemon master. The function has 4 parameters. The first parameter is `social_network` (as clarified above), the second one is the id of the person that you will recommend friends; this parameter is an integer. The third parameter is the distance which is an integer. You should recommend friends to the given pokemon trainer who is not farther than the given distance. The one constraint is you should recommend friends for the given pokemon trainer but these recommendations should be benefical for him. Pokemon trainers usually want to be friends with the pokemon trainers who has got a specific pokemon. The name of the pokemon is given as the 4th parameter and it is a string. After you construct the recommendation list you should eliminate the pokemon trainers who does not have the given pokemon. Also you should be careful to not recommend people that are already friends.

  As an example if you are given the above social network as social network (the original one given at Social Network section), 7 as trainerid 5 as distance and "pikachu" as pokemon name you should return [1]. If the trainerid were 3 and the distance were 5 then you should return [1,7]. Remember that the lists you return should be sorted. So, returning [7,1] is a wrong answer.

- `catch_pokemon social_network pokemon_pairs` : In this function you will help Professor Oak to program his new idea. He thinks that some pokemons are related, meaning that a trainer who has one pokemon can catch a related pokemon. For example if a trainer has slowpoke, the Professor Oak thinks that trainer will catch a shellder sooner or later. In this function you will implement the Professor Oak's idea. catch-pokemon function will have two parameters. The first one is `social_network` (as clarified above) and the second parameter is a list of pokemon pairs like [("slowpoke","shellder"),("merill","venonat"),("staryu","squirtle")]. In this function your job is to return a modified `social_network` in which if a trainer has one of the pairs, the other pokemon in the pair should be added to that trainers pokemon list. All of the pokemon lists of the trainers are given sorted to you also after this additions they should be sorted. (From a to z).

  For example if you given the above social network and the above pairs (pairs given in this function) then Misty's pokemon list will be ["squirtle","staryu"] and Tracey's pokemon list will be ["merill","venonat"] and the Ash'es pokemon list will be ["bulbasaur" , "charmender" , "pikachu" , "squirtle", "staryu"] other trainers list should not change.

- `get_statistics social_network gender (above_or_below,age)`: This time Professor Ivy needs your help. In this function you will have three parameters as input and you will return a list of trainer ids. Professor Ivy is making a new research and for this reason needs the trainer ids of some age intervals and genders's. The first parameter is again `social_network`, the second parameter is gender, meaning that it will either be "female" or "male". The third parameter which is a pair gives information about the age interval. `above_or_below` is an string and can have two values either "above" or "below" and the age is an integer in the interval of 0 and 100. You should return a list of all trainer id's satisfying the given conditions.

For example if the gender is given as "male" and the pair is given as ("below",14) you should return the list of trainer id's that are "male" and has age 0 to 13. In our `social_network` example this query would return, [1,3,7]. Because Ash, Gary and Ritchie only satisfies this query. We don't have Misty because her gender is "female" and we don't have Brock and Tracey because they are too old for this query. Also the list you return here must be sorted (in increasing order) as I write in the example.

4 Functions to be provided to you

In order to make your job easier we will provide you a " help.hs " file. This file contains some functions which you are free to copy them to your own homework file and use. There are many functions in this hs file but only 3 of them are important for you. Other functions are helper functions of the main functions provided to you. You won't be sending two files to us. So when you want to use one of these functions, just copy and paste the definitions and use as you wish.

1. **dijkstra graph baslangic bitis**
   This function takes a graph a starting node and an ending node. The result it gives is the distance between the nodes baslangic and bitis. The graph parameter is basically the second part of your social network (friendship relationships).

   As an example dijkstra [(1,2) , (1,3) , (1,4) , (2,3) , (3,6) , (3,9) , (6,7) , (7,9)] 1 9 returns 2. Not 4 or 3.
   As another example dijkstra [(1,2) , (1,3) , (1,4) , (1,7) , (2,3) , (3,6) , (6,7) , (7,9)] 4 9 return 3. Not 5.

2. `insertion_sort_lst list_to_be_sorted new_list`
   In this function as first parameter you should provide a list containing Integers or Strings. As second parameter you should provide an empty list. The function returns a list which is sorted. (It uses insertion sort algorithm.).

   As an example `insertion_sort_lst` [3,1,10,19,-4] [ ] will return [-4,1,3,10,19].
   Or `insertion_sort_lst` ["Psyduck","Pikachu","Raichu"] [ ] will return ["Pikachu","Psyduck","Raichu"]

3. `insertion_sort_tuple tuple_list_to_be_sorted new_list`
   In this function as first parameter you should provide a list containing an Integer tuple (like your friendship relationship list). As second parameter an empty list. The function returns a list of tuples which is sorted. (It uses insertion sort algorithm.).

   As an example insertion_sort_tuple [(1,2),(9,11),(3,7),(8,5),(3,1)] [ ]
   will return [(1,2),(3,1),(3,7),(8,5),(9,11)] .

5 Regulations

1. **Programming Language:** Haskell
2. **Late Submission:** You have 3 days out of 10 (total lates for all homeworks) for the late submission.
3. **Cheating: We have zero tolerance policy for cheating.** People involved in cheating will receive zero from all homework (hence get NA from this course).
4. **Newsgroup:** You must follow the newsgroup (news.ceng.metu.edu.tr) for discussions and possible updates on a daily basis.

5. **Evaluation:** Your program will be evaluated automatically using black-box technique so make sure to obey the specifications. Your codes will also be visually investigated and code analysis tool will be used to catch similarities between submissions. For that reason, never share your code with others.

6 Submission

Submission will be done via COW. Create a tar document named hw2.tar.gz which contains a single file named hw2.hs that contains your code.
Note: The submitted archive should not contain any directories! Failing to obey this specification will cause deduction in your grades.