Sabancı University

Faculty of Engineering and Natural Sciences
CS204 Advanced Programming
Spring 2019

Homework 3 – Map Representation & Manipulation with Linked Lists

Due: 20/03/2019, Wednesday, 21:00

PLEASE NOTE:

Your program should be a robust one such that you have to consider all relevant programmer mistakes and extreme cases; you are expected to take actions accordingly!

You can NOT collaborate with your friends and discuss solutions. You have to write down the code on your own. Plagiarism will not be tolerated!

Introduction

In this homework, you are asked to implement a map representation of Game of Thrones Westeros. In the map, you must use a 2-dimensional *Linked List* structure to store information about *Houses* and their *Neighbors*. For each house, the neighboring houses must be stored in a *Linked List*. Moreover, these linked lists are also connected that makes the data structure you will use a hybrid and 2-dimensional one. Your program will be able to get data from a file and perform some other operations. Details will be explained in the subsequent sections of this homework document.

The Program Flow

The program gets input data from a text file. For this, it asks for the name of the input file. If the user enters an invalid file name, the program asks the name of the input file until a correct file name is given and file is opened successfully. The information in the file is organized as rows. In each row, the first entry is the name of the house (as a string), the second entry is the name of the neighboring house (as a string). The entries are separated by arbitrary number of blank characters. See Figure 1 below for a sample input file.

The map will be stored in a data structure that is made of several linked lists. Your program should read each line and modify the data structure. Details on the data structure and how to modify it will be given in the upcoming parts of the document.

You may assume that your input file is given in the correct format which means you do not need to perform input checks.

```
Tyrell      Lanister
Tyrell   Martell
Stark     Tully
Stark   Arryn
Tully            Arryn
Tully  Greyjoy
Lanister   Tully
Greyjoy    Stark
Targaryen Tully
Targaryen    Tyrell
Tully Tyrell
Baratheon Tyrell
Baratheon  Targaryen
Baratheon            Martell
Lanister  Greyjoy
```

Figure 1. Sample input file

After loading and correctly displaying the map information, the program prompts a menu that contains a list of operations. The menu provides 3 different options and each option can be chosen in any order. When an option is entered, the corresponding operation is performed and then the menu is displayed again. This continues until the user chooses the option to end the program. The menu options and the corresponding operations are explained below.

1. **Annexation! War for the iron throne continues…:**

In this option, the inputs are taken from the keyboard. The user is going to enter two house names that exists in the map (in case of non-existence an error message must be displayed). The first house name entered is going to conquer the second one, which means, now the first house becomes neighbors with the second houses neighbors (same neighbors case must be thought carefully). Also, since now the second house will not exist in the map, entries belonging to that house must be deleted from the linked list after annexation. Please remark that this option has some special cases to be considered such as being neighbors, having same neighbors, etc. Algorithms for necessary updates must be thought and coded carefully. Moreover, you have to consider the case where both house names are entered the same (of course a house cannot conquer itself) and the case where a house tries to conquer a non-neighbor (this should not be allowed).

2. **Tyrion must travel, do a path check for the famous imp:**

In this option, the inputs are taken from the keyboard. This option is to check the correctness of a path in the map. The user is asked to enter the names of the houses that Tyrion is going to go through in the correct order. If such a path exists, your program will output that the path is found. If the path cannot be found, your program must alert the user that Tyrion cannot start such a journey. Please remark that the path must be processed in pairs. For example, if the path entered is "Stark Arryn Tyrell Martell", then Stark - Arryn, Arryn - Tyrell, and Tyrell - Martell must be all neighbors to qualify this path as a valid one. As the output, you have to display results of pairwise checks in addition to ultimate path check result. Having the same house name consecutively in the path is not an error, but while processing this pair, a special message saying that the traveler is in the same house would be good to have. Needless to say that these neighboring relationships must be checked through the data structure.

3. Exit:
Program terminates when the user chooses this option.

If another option value is entered by the user, since it is invalid, the menu will be displayed again with an error message.

The Data Structure to be Used

In this homework, you must represent houses as a *linked list*, and represent the neighbors of each house as another *linked list* starting from the corresponding house node. This way, you will end up a two dimensional data structure. Figure 2 depicts the data structure that you will use in this homework.

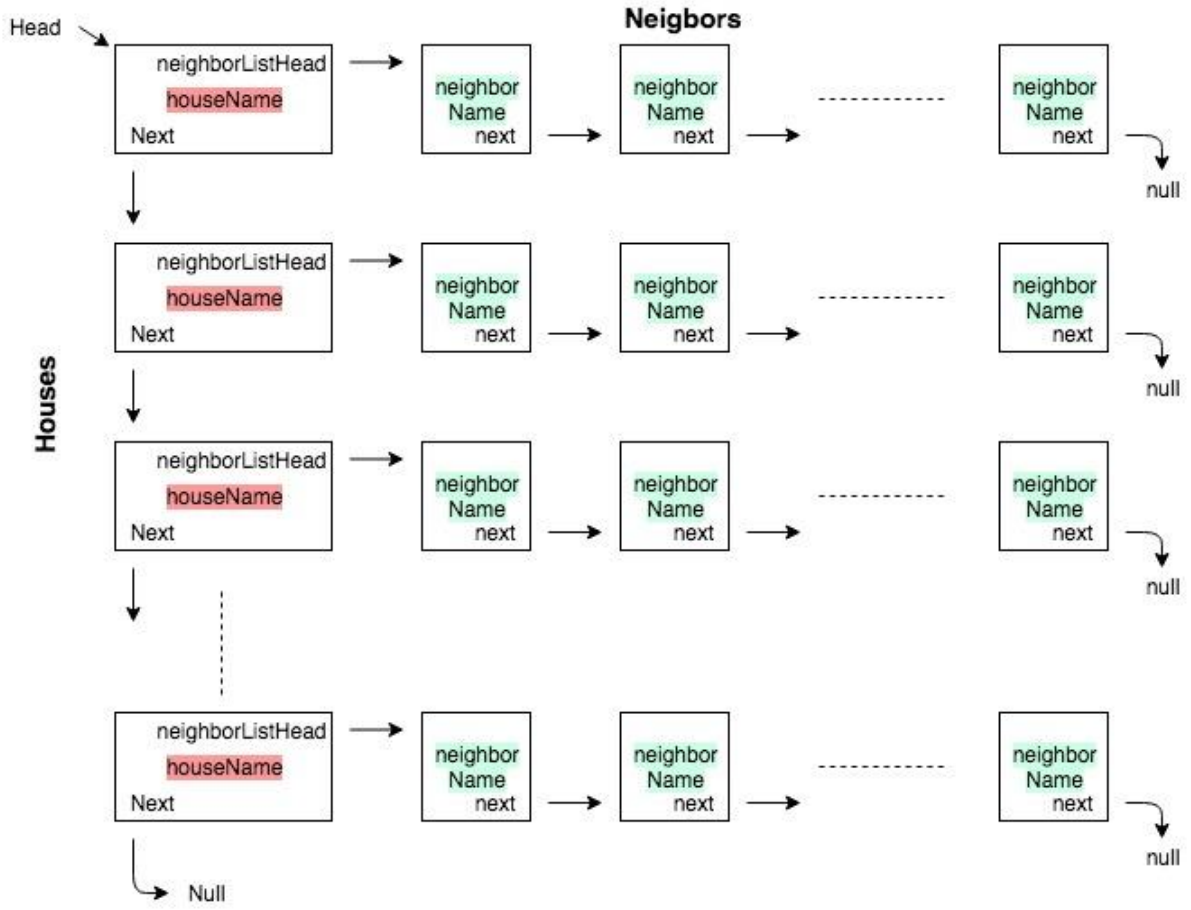

Figure 2. The data structure

To help the implementation of this data structure, you are given two different node types: one for the linked list that holds the neighbors, and one for the linked list to hold the houses.

The node type for the neighbor is a regular linked list node with one `next` pointer and one data field, which is a string for the name of the neighboring house. Partial node `struct` for the neighbor node is given below.

```
struct neighborNode
{
	string neighborName;
	neighborNode * next;

	// constructors come here
};
```

The node type to hold house information is the one which has two pointers. It has one `next` and one `neighborListHead` pointer. `next` is of type `houseNode` pointer, which is used to create Houses linked list, and `neighborListHead` is of type `neighborNode` pointer, which points to the start of the neighbor linked list of the corresponding house. In addition, this node type also keeps a string for the name of the house. Partial node `struct` for the house node is given below.

```
struct houseNode
{
	string houseName;
	neighborNode * neighborListHead;
	houseNode * next;

	// constructors come here
};
```

You are requested to design and implement a class for this special hybrid and 2-dimensional *linked list*. Name this class as `linkedlist`, keep the `head` as a private data member and perform all operations via the member functions of this class. Partial class definition is given below. You can add some other helper/member functions, if you need to do so. You can copy/paste this class definition and structs given above in a header file. However, you have to implement the member functions in another .cpp file (other than the one includes main) and use them in your main program.

```
class linkedlist
{
public:
	linkedlist();
	void addHouseNode(string hname);
	void addNeighborNode (string hname, string nname);
	void printAll();
	bool hnExists(string hname);
	bool nnExists(string hname, string nname);
	void deleteAll();  //to deallocate all dynamic data
	//other member functions…

private:
	houseNode * head;
	// any helper functions you see necessary
	// ...
};
```

addHouseNode(string hname): This member function is going to add a houseNode, to the end of the houses linked list vertically with house name given as parameter. There must not be two house nodes with the same house name in the list; this control can be done before calling this function in main or in this function depending on your design.

addNeighborNode (string hname, string nname): This member function is going to add a neighborNode, with neighbor name nname, to the end of the neighbor list of a houseNode with house name hname, horizontally. A particular neighbor name can appear at most once in the neighbor list of a house; this control can be done before calling this function in main or in this function depending on your design.

printAll(): This member function prints the entire list structure (i.e. the map) to the screen as shown in the sample runs.

hnExists(string hname): This function returns true if a houseNode with the parameter house name (hname) exists in the data structure. Otherwise returns false.

nnExists(string hname, string nname): This function returns true if a neighborNode with the parameter neighbor name (nname) exists in the neighbor list of houseNode with house name hname. Otherwise returns false. In other words, this function returns true if nname is a neighbor of hname; returns false otherwise.

deleteAll(): At the end of your program the program will exit after calling this function which deletes the entire linked list structure.

Other than these member functions, you can add more (for example for annexation purposes).

<u>No</u> other multiple data containers (built-in array, dynamic array, vector, matrix, 2D array, etc.) can be used.
Moreover, you cannot use extra files to avoid linked list usage and you cannot process the input map file more than once; the input map file will be read only once and the map data structure will be created accordingly. After that all of the operations will be performed over that data structure.

Addition and Deletion of Data to Data Structure; Handling Redundant Data

In the Houses list, there cannot be multiple nodes with the same house name. When a new record is about to be added to the map, you must first check whether given house name exists in the map. If the name does not exist, then you need to append a new house node to the end of the Houses list. After the insertion, the neighbor data is going to be added to the corresponding neighbor list of that house as the first element. If there is already a node with the given name in the Houses list, i.e. if the house is already in the map, then you must append the neighbor information to the end of the neighbors list of this existing house. You do not need to store houses and neighbors in sorted fashion.

For each house pair read from the file, data addition must be done to reflect the fact that both of the houses are neighbors of each other. For example, if the input line is "Tyrell Lanister", you have to add a house node for "Tyrell" and a neighbor node "Lanister" for it. And you also have to add another house node for "Lanister" and a neighbor node "Tyrell" for it.

For all house pairs read from the file, you have to display an appropriate message as exemplified in the sample runs section below (if the neighbor pair is appeared first, a "successful processing" message; if the pair has appeared before, a "redundant information" message). Appearance of a house pair more than once can be in the same of reverse orders; both cases are redundant. Appearance of a house pair more than once must be checked via the existence checks

in the data structure (you will implement necessary member functions for that), not by remembering or processing file data directly.

Whenever a house node needs to be deleted, you must deallocate all the neighbor nodes belonging to the house before deallocating the house node itself.

Moreover, you must deallocate all dynamic memory before your program terminates. You have to perform this by implementing and calling `deleteAll` member function of the class.

A sample data structure to be generated after processing the data file given in Figure 1 is depicted in Figure 3. Please do not forget that this is just an example, your program must work with any map, any house names and any neighboring relationships.

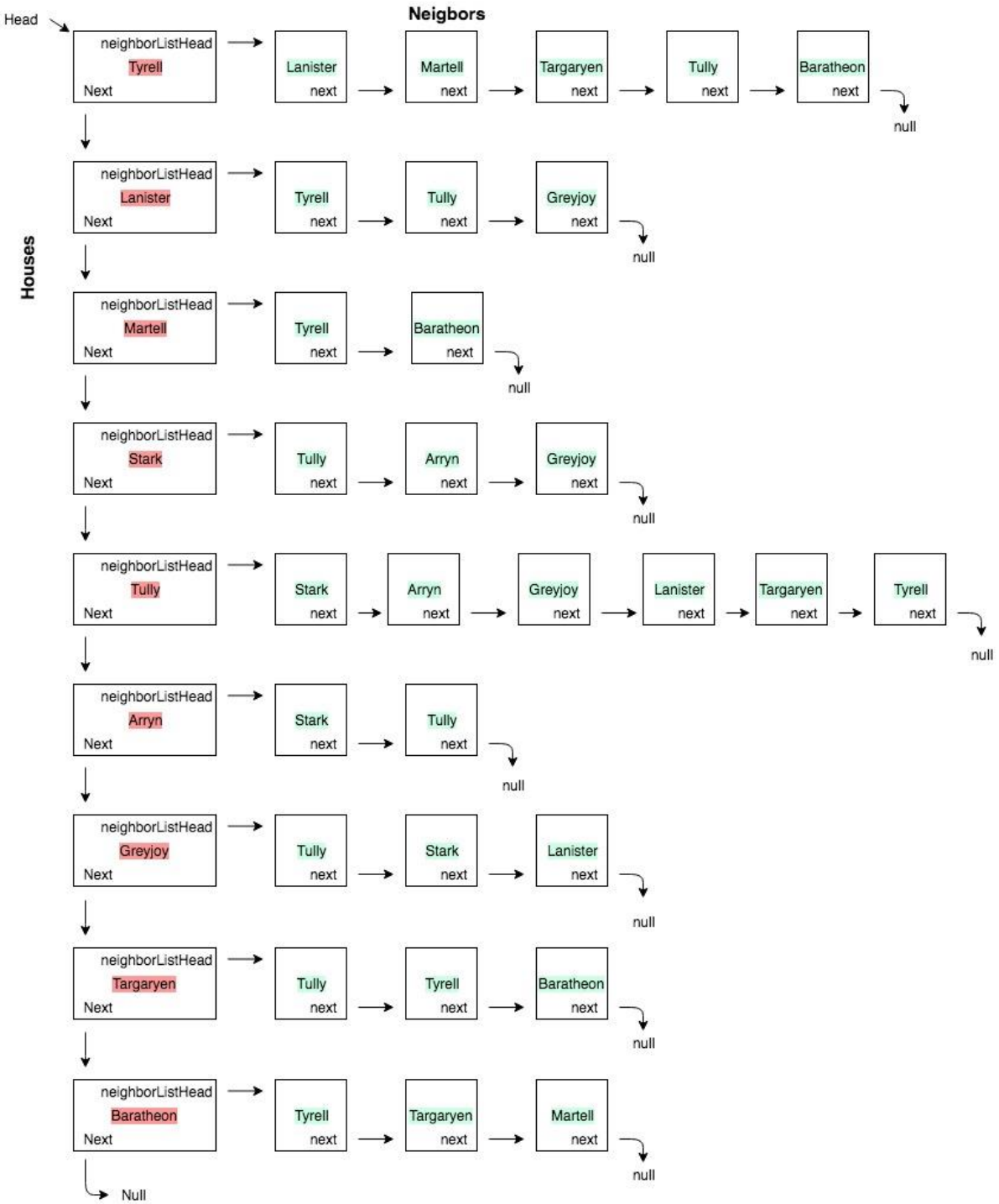

Figure 3. A sample data structure

Sample Runs

Sample runs are given below, but these are not comprehensive, therefore you have to consider **all possible cases** to get full mark.

The sample input files are provided in the .zip package of this homework.

All of the sample runs are created by using westeros.txt provided in the zip file. However, please keep in mind that we will use another file (or files) with different house names and neighboring relationships in grading. Thus, please do not make any assumptions about the house names and neighboring relationships.

Sample Run 1:

```
Please enter a file name.
westeros.txt
Successfully opened file westeros.txt

######## TOPOLOGY ########

Processing Tyrell, Lanister
Topology information successfully added.

Processing Tyrell, Martell
Topology information successfully added.

Processing Stark, Tully
Topology information successfully added.

Processing Stark, Arryn
Topology information successfully added.

Processing Tully, Arryn
Topology information successfully added.

Processing Lanister, Tyrell
Redundant information! An entry with Lanister and Tyrell is already processed.

Processing Tully, Greyjoy
Topology information successfully added.

Processing Lanister, Tully
Topology information successfully added.

Processing Greyjoy, Stark
Topology information successfully added.

Processing Targaryen, Tully
Topology information successfully added.

Processing Targaryen, Tyrell
Topology information successfully added.
```

Processing Tully, Tyrell
Topology information successfully added.

Processing Stark, Greyjoy
Redundant information! An entry with Stark and Greyjoy is already processed.

Processing Baratheon, Tyrell
Topology information successfully added.

Processing Baratheon, Targaryen
Topology information successfully added.

Processing Greyjoy, Stark
Redundant information! An entry with Greyjoy and Stark is already processed.

Processing Baratheon, Martell
Topology information successfully added.

Processing Lanister, Greyjoy
Topology information successfully added.

######## MAP ########

Tyrell: Lanister, Martell, Targaryen, Tully, Baratheon,
Lanister: Tyrell, Tully, Greyjoy,
Martell: Tyrell, Baratheon,
Stark: Tully, Arryn, Greyjoy,
Tully: Stark, Arryn, Greyjoy, Lanister, Targaryen, Tyrell,
Arryn: Stark, Tully,
Greyjoy: Tully, Stark, Lanister,
Targaryen: Tully, Tyrell, Baratheon,
Baratheon: Tyrell, Targaryen, Martell,

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
2
Please enter a path. (ex: Tyrell Martell Tully)
Tyrell Lanister Tully Stark
Path found between Tyrell and Lanister
Path found between Lanister and Tully
Path found between Tully and Stark
Path search succeeded.

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
2
Please enter a path. (ex: Tyrell Martell Tully)

Stark Stark Greyjoy
You are already in Stark
Path found between Stark and Greyjoy
Path search succeeded.

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
5
Invalid option please select from the menu.

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
2
Please enter a path. (ex: Tyrell Martell Tully)
Dorne Baratheon
Dorne does not exist in the map.
Path search failed!

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
2
Please enter a path. (ex: Tyrell Martell Tully)
Targaryen Tyrell Lanister Baratheon
Path found between Targaryen and Tyrell
Path found between Tyrell and Lanister
Baratheon is not a neighbor of Lanister
Path search failed!

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
3
List deleted and program ended.

Sample Run 2:

Please enter a file name.
westeros.txt
Successfully opened file westeros.txt

######## TOPOLOGY ########

Processing Tyrell, Lanister
Topology information successfully added.

Processing Tyrell, Martell

Topology information successfully added.

Processing Stark, Tully
Topology information successfully added.

Processing Stark, Arryn
Topology information successfully added.

Processing Tully, Arryn
Topology information successfully added.

Processing Lanister, Tyrell
Redundant information! An entry with Lanister and Tyrell is already processed.

Processing Tully, Greyjoy
Topology information successfully added.

Processing Lanister, Tully
Topology information successfully added.

Processing Greyjoy, Stark
Topology information successfully added.

Processing Targaryen, Tully
Topology information successfully added.

Processing Targaryen, Tyrell
Topology information successfully added.

Processing Tully, Tyrell
Topology information successfully added.

Processing Stark, Greyjoy
Redundant information! An entry with Stark and Greyjoy is already processed.

Processing Baratheon, Tyrell
Topology information successfully added.

Processing Baratheon, Targaryen
Topology information successfully added.

Processing Greyjoy, Stark
Redundant information! An entry with Greyjoy and Stark is already processed.

Processing Baratheon, Martell
Topology information successfully added.

Processing Lanister, Greyjoy
Topology information successfully added.

######## MAP ########

Tyrell: Lanister, Martell, Targaryen, Tully, Baratheon,

Lanister: Tyrell, Tully, Greyjoy,
Martell: Tyrell, Baratheon,
Stark: Tully, Arryn, Greyjoy,
Tully: Stark, Arryn, Greyjoy, Lanister, Targaryen, Tyrell,
Arryn: Stark, Tully,
Greyjoy: Tully, Stark, Lanister,
Targaryen: Tully, Tyrell, Baratheon,
Baratheon: Tyrell, Targaryen, Martell,

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
1
Please enter two houses first the annexer and the annexed. (ex: Tyrell Stark).
Lanister Arryn
A house can only conquer a neighboring house! Arryn is not a neighbor of Lanister.
Update failed.

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
1
Please enter two houses first the annexer and the annexed. (ex: Tyrell Stark).
Stark Stark
A House cannot conquer itself!
Update failed.

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
1
Please enter two houses first the annexer and the annexed. (ex: Tyrell Stark).
Dorne Tully
Dorne does not exist!
Update failed.

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
1
Please enter two houses first the annexer and the annexed. (ex: Tyrell Stark).
Tully Bravos
Bravos does not exist!
Update failed.

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit

1
Please enter two houses first the annexer and the annexed. (ex: Tyrell Stark).
Lanister Tyrell
Lanister conquered Tyrell!
######## MAP ########
Lanister: Tully, Greyjoy, Martell, Targaryen, Baratheon,
Martell: Lanister, Baratheon,
Stark: Tully, Arryn, Greyjoy,
Tully: Stark, Arryn, Greyjoy, Lanister, Targaryen,
Arryn: Stark, Tully,
Greyjoy: Tully, Stark, Lanister,
Targaryen: Tully, Lanister, Baratheon,
Baratheon: Lanister, Targaryen, Martell,

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
1
Please enter two houses first the annexer and the annexed. (ex: Tyrell Stark).
Lanister Tully
Lanister conquered Tully!
######## MAP ########
Lanister: Greyjoy, Martell, Targaryen, Baratheon, Stark, Arryn,
Martell: Lanister, Baratheon,
Stark: Lanister, Arryn, Greyjoy,
Arryn: Stark, Lanister,
Greyjoy: Stark, Lanister,
Targaryen: Lanister, Baratheon,
Baratheon: Lanister, Targaryen, Martell,

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
1
Please enter two houses first the annexer and the annexed. (ex: Tyrell Stark).
Lanister Baratheon
Lanister conquered Baratheon!
######## MAP ########
Lanister: Greyjoy, Martell, Targaryen, Stark, Arryn,
Martell: Lanister,
Stark: Lanister, Arryn, Greyjoy,
Arryn: Stark, Lanister,
Greyjoy: Stark, Lanister,
Targaryen: Lanister,

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
1
Please enter two houses first the annexer and the annexed. (ex: Tyrell Stark).

Lanister Arryn
Lanister conquered Arryn!
######## MAP ########
Lanister: Greyjoy, Martell, Targaryen, Stark,
Martell: Lanister,
Stark: Lanister, Greyjoy,
Greyjoy: Stark, Lanister,
Targaryen: Lanister,

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
1
Please enter two houses first the annexer and the annexed. (ex: Tyrell Stark).
Greyjoy Stark
Greyjoy conquered Stark!
######## MAP ########
Lanister: Greyjoy, Martell, Targaryen,
Martell: Lanister,
Greyjoy: Lanister,
Targaryen: Lanister,

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
1
Please enter two houses first the annexer and the annexed. (ex: Tyrell Stark).
Greyjoy Martell
A house can only conquer a neighboring house! Martell is not a neighbor of Greyjoy.
Update failed.

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
1
Please enter two houses first the annexer and the annexed. (ex: Tyrell Stark).
Lanister Martell
Lanister conquered Martell!
######## MAP ########
Lanister: Greyjoy, Targaryen,
Greyjoy: Lanister,
Targaryen: Lanister,

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
1
Please enter two houses first the annexer and the annexed. (ex: Tyrell Stark).
Lanister Greyjoy
Lanister conquered Greyjoy!

######## MAP ########
Lanister: Targaryen,
Targaryen: Lanister,

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
1
Please enter two houses first the annexer and the annexed. (ex: Tyrell Stark).
Targaryen Lanister
Targaryen conquered Lanister!
######## MAP ########
Targaryen:

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
3
List deleted and program ended.

Sample Run 3:

Please enter a file name.
westeros.txt
Successfully opened file westeros.txt

######## TOPOLOGY ########

Processing Tyrell, Lanister
Topology information successfully added.

Processing Tyrell, Martell
Topology information successfully added.

Processing Stark, Tully
Topology information successfully added.

Processing Stark, Arryn
Topology information successfully added.

Processing Tully, Arryn
Topology information successfully added.

Processing Lanister, Tyrell
Redundant information! An entry with Lanister and Tyrell is already processed.

Processing Tully, Greyjoy
Topology information successfully added.

Processing Lanister, Tully

Topology information successfully added.

Processing Greyjoy, Stark
Topology information successfully added.

Processing Targaryen, Tully
Topology information successfully added.

Processing Targaryen, Tyrell
Topology information successfully added.

Processing Tully, Tyrell
Topology information successfully added.

Processing Stark, Greyjoy
Redundant information! An entry with Stark and Greyjoy is already processed.

Processing Baratheon, Tyrell
Topology information successfully added.

Processing Baratheon, Targaryen
Topology information successfully added.

Processing Greyjoy, Stark
Redundant information! An entry with Greyjoy and Stark is already processed.

Processing Baratheon, Martell
Topology information successfully added.

Processing Lanister, Greyjoy
Topology information successfully added.

######## MAP ########

Tyrell: Lanister, Martell, Targaryen, Tully, Baratheon,
Lanister: Tyrell, Tully, Greyjoy,
Martell: Tyrell, Baratheon,
Stark: Tully, Arryn, Greyjoy,
Tully: Stark, Arryn, Greyjoy, Lanister, Targaryen, Tyrell,
Arryn: Stark, Tully,
Greyjoy: Tully, Stark, Lanister,
Targaryen: Tully, Tyrell, Baratheon,
Baratheon: Tyrell, Targaryen, Martell,

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
2
Please enter a path. (ex: Tyrell Martell Tully)
Targaryen Baratheon Lanister
Path found between Targaryen and Baratheon

Lanister is not a neighbor of Baratheon
Path search failed!

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
1
Please enter two houses first the annexer and the annexed. (ex: Tyrell Stark).
Lanister Tyrell
Lanister conquered Tyrell!
######## MAP ########
Lanister: Tully, Greyjoy, Martell, Targaryen, Baratheon,
Martell: Lanister, Baratheon,
Stark: Tully, Arryn, Greyjoy,
Tully: Stark, Arryn, Greyjoy, Lanister, Targaryen,
Arryn: Stark, Tully,
Greyjoy: Tully, Stark, Lanister,
Targaryen: Tully, Lanister, Baratheon,
Baratheon: Lanister, Targaryen, Martell,

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
2
Please enter a path. (ex: Tyrell Martell Tully)
Targaryen Baratheon Lanister
Path found between Targaryen and Baratheon
Path found between Baratheon and Lanister
Path search succeeded.

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
2
Please enter a path. (ex: Tyrell Martell Tully)
Stark Tuly Targaryen Martell
Tuly is not a neighbor of Stark
Path search failed!

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
1
Please enter two houses first the annexer and the annexed. (ex: Tyrell Stark).
Targaryen tully
tully does not exist!
Update failed.

Please select an option [1-2-3].

1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
1
Please enter two houses first the annexer and the annexed. (ex: Tyrell Stark).
Targaryen Tully
Targaryen conquered Tully!
######## MAP ########
Lanister: Greyjoy, Martell, Targaryen, Baratheon,
Martell: Lanister, Baratheon,
Stark: Targaryen, Arryn, Greyjoy,
Arryn: Stark, Targaryen,
Greyjoy: Targaryen, Stark, Lanister,
Targaryen: Lanister, Baratheon, Stark, Arryn, Greyjoy,
Baratheon: Lanister, Targaryen, Martell,

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
4
Invalid option please select from the menu.

Please select an option [1-2-3].
1. Annexation! War for the iron throne continues...
2. Tyrion must travel, do a path check for the famous imp.
3. Exit
3
List deleted and program ended.

Some Important Rules

In order to get a full credit, your programs must be efficient and well presented, presence of any redundant computation or bad indentation, or missing, irrelevant comments are going to decrease your grades. You also have to use understandable identifier names, informative introduction and prompts. Modularity is also important; you have to use functions wherever needed and appropriate. Since using classes is mandated in this homework, a proper object oriented design and implementation will also be considered in grading.

Since you will use dynamic memory allocation in this homework, it is very crucial to properly manage the allocated area and return the deleted parts to the heap whenever appropriate. Inefficient use of memory may reduce your grade.

When we grade your homework we pay attention to these issues. Moreover, in order to observe the real performance of your codes, we may run your programs in *Release* mode and **we may test your programs with very large test cases**. Of course, your program should work in *Debug* mode as well.

You are allowed to use sample codes shared with the class by the instructor and TAs. However, you cannot start with an existing .cpp or .h file directly and update it; you have start with an empty file. Only the necessary parts of the shared code files can be used and these parts must be clearly marked in your homework by putting comments like the following. Even if you take

a piece of code and update it slightly, you have to put a similar marking (by adding "`and updated`" to the comments below.

```
/* Begin: code taken from ptrfunc.cpp */

…

/* End: code taken from ptrfunc.cpp */
```

What and where to submit (PLEASE READ, IMPORTANT)

You should prepare (or at least test) your program using MS Visual Studio 2012 C++. We will use the standard C++ compiler and libraries of the abovementioned platform while testing your homework. It'd be a good idea to write your name and last name in the program (as a comment line of course).

Submissions guidelines are below. Some parts of the grading process are automatic. Students are expected to strictly follow these guidelines in order to have a smooth grading process. If you do not follow these guidelines, depending on the severity of the problem created during the grading process, 5 or more penalty points are to be deducted from the grade. Name your solution, project, cpp file that contains your main program using the following convention (the necessary file extensions such as .sln, .cpp, etc, are to be added to it):

"SUCourseUserName_YourLastname_YourName_HWnumber"

Your SUCourse user name is actually your SUNet user name which is used for checking sabanciuniv e-mails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name. For example, if your SUCourse user name is cago, name is Çağlayan, and last name is Özbugsızkodyazaroğlu, then the file name must be:

Cago_Ozbugsizkodyazaroglu_Caglayan_hw3

In some homework assignments, you may need to have more than one .cpp or .h files to submit. In this case add informative phrases after the hw number. However, do not add any other character or phrase to the file names.

Now let us explain which files will be included in the submitted package. Visual Studio 2012 will create two *debug* folders, one for the solution and the other one for the project. You should delete these two *debug* folders. Moreover, if you have run your program in release mode, Visual Studio may create *release* folders; you should delete these as well. Apart from these, Visual Studio 2012 creates a file extension of *.sdf*; you will also delete this file. The remaining content of your solution folder is to be submitted after compression. Compress your solution and project folders using WINZIP or WINRAR programs. Please use "zip" compression. "rar" or another compression mechanism is NOT allowed. Our homework processing system works only with zip files. Therefore, make sure that the resulting compressed file has a zip extension. Check that your compressed file opens up correctly and it contains all of the solution, project and source code files that belong to the latest version of your homework. Especially double-check that the zip file contains your cpp and (if any) header files that you wrote for the homework.

Moreover, we strongly recommend you to check whether your zip file will open up and run correctly. To do so, unzip your zip file to another location. Then, open your solution by clicking the file that has a file extension of .sln. Clean, build and run the solution; if there is no problem, you could submit your zip file. Please note that the deleted files/folders may be regenerated after you build and run your program; this is normal, but do not include them in the submitted zip file.

You will receive no credits if your compressed zip file does not expand or it does not contain the correct files. The naming convention of the zip file is the same. The name of the zip file should be as follows:

SUCourseUserName_YourLastname_YourName_HWnumber.zip

For example, zubzipler_Zipleroglu_Zubeyir_hw3.zip is a valid name, but

Hw3_hoz_HasanOz.zip, HasanOzHoz.zip

are **NOT** valid names.

Submit via SUCourse ONLY! You will receive no credits if you submit by other means (e-mail, paper, etc.).

Successful submission is one of the requirements of the homework. If, for some reason, you cannot successfully submit your homework and we cannot grade it, your grade will be 0.

Good Luck!

Albert Levi, Taha Atahan Akyıldız