Final Project: Modelling the Heredity of Dominant Genetic Diseases

Minh Duc Hoang, Katherine Liang, Emily So

August 2, 2020

1 Project Description

1.1 Motivation

An area of great focus in medicine currently is studying the inheritance of genetic diseases among lineages, and as three students passionate about medicine, we wanted to create an abstract data type that is able to model real-life heredity of genetic diseases. In the field of genetic disease, individuals can undergo genetic testing in which their entire genome is sequenced to identify gene mutations that can harbor different forms of abnormalities. Depending on the nature of the genetic mutation, this can have a great effect on one's children. Genetic diseases can be dominant (the child only needs one copy of the gene from one parent for the disease to manifest) or recessive (child must have two copies of the abnormal gene from both parents to have the disease (What are the different ways in which a genetic condition can be inherited? - Genetics Home Reference - NIH). For many of these diseases, there is currently no cure, so by modelling their heredity using an ADT, we are hoping that more people can be encouraged to apply data structures when researching genetic diseases.

1.2 Problem Statement

For this project, we are specifically interested in dominant genetic diseases, such as Huntington's Disease. We wish to predict whether or not an individual will have a dominant genetic disease given the health information of all members in any generation of the individual's ancestry.

2 ADT: Tree

2.1 Data

We will use the tree ADT to model the problem at hand. The data required for this ADT is a set of nodes, where:

- each node represents one member of the family.
- each node contains the health status of that family member with respect to a particular dominant genetic disease.
- all nodes in the same level belong to the same generation.
- the root of the tree is the individual in question, his/her parents' nodes are in the second layer, his/her grandparents' nodes are in the third layer, and so on.

2.2 Operations

The tree ADT will support the following operations:

- INSERT(S, j): This adds family member j (along with its genetic information) into the family tree S. Depending on member j's genetic condition, performing INSERT(S, j) may result in updates to other members' genetic status. This operation is similar to a common occurrence in genealogy, in which "missing parts" of a family tree continue to be filled in as more information comes to light.
- DELETE(S, j): This deletes family member j (along with its genetic information) from the family tree S. Depending on member j's genetic condition, performing DELETE(S, j) may result in updates to other members' genetic status. The application of this is when certain family members are found to be unrelated to other family members of the family tree and should be removed to better determine the genetic status of the individual at the root node.
- RETURN_ROOT(S): This is used to determine whether or not the individual we are interested in (as represented by the root) has a given dominant genetic disease.
- STATUS(S, j): This is used to determine the genetic condition of member j.
- MAXIMUM(S): This is used to find one of the oldest members of the family that was affected by a certain dominant genetic disease. In the real-life context, this helps determine the origin of the genetic disease.
- MINIMUM(S): This is used to identify the youngest member of the family that is affected by the dominant genetic disease. In the real-life context, this helps determine which generation the genetic disease stopped affecting a specific family.