Thesis

# Introduction

Puzzle – other man – “Brothers and sisters I have none, but this man’s father is my father’s son”.

This puzzle’s solution is “my son”. However in this particular situation, with greek mythology, it could be something far more obscure – divine father, disputed father.

# Graph data formats considered

## Straight JSON

## GraphLib

Not directly a data representation format, however is used to store the information about nodes and edges.

Contextualises the information into nodes and edges, saves having to generate this ourselves. Has a lot of implicit functions already provided for finding edges and nodes, and formatting them etc.

However does not create an actual database – we generally use an existing database of information to use GraphLib. Difficult to understand the complexities of this kind of data.

However upon investigation this seems to be the best method of storing our information. A combination of JSON converted information from the original CSV file, to graphlib objects.

## GEDCOM

The standard format for genealogical data.

(GEnealogical Data COMmunications)

Developed by the Church of Jesus Christ and Latter-Day Saints. Used for recording baptisms(?)

Text file.

**Pros:**

Recognised standard for genealogical data

**Cons:**

# Types of graph layout software / graph visualisation tools considered

## GraphViz

## DagreJS asnd D3.js + SVG?

Really good

Allows for interactable elements – good for linking back to the source content and highlighting the individual unusual relationships for focus.

Unfortunately is impossible / difficult to format the existing layout algorithm. And also genealogical graphing is not accessible.

## Canvas.js

## Legacy 9

Accessible for different types of data input, for example can import GEDCOM text files.

**Pros:**

* Can input the number of generations wanting to show
* Automatically displays the information in genealogical format
* Can determine whether to show ancestry or descendants
* Can choose which entity is the focus entity (usually the left-most one)
* Imports existing file data such as GEDCOM
* Allows for different genealogical (family tree) formats, e.g. traditional nodes and edges, or ancestry fan
* Allows for different layouts depending on page size (e.g. horizontal and vertical family tree representation)
* Immediately obvous is a genealogy chart

**Cons:**

* Can’t show descendants and ancestry in the same graph?
* Costs money
* Layout algorithms not directly accessible and not able to be converted (no API) for putting that data on the screen – generate dynamically, is not interactable
* Assume every entity is human (but can name them “void”)
* Can handle brother and sister relationships, but not inter-generational relationships (integral to greek mythology)
* Can’t list alternative parents (‘other parents’) and cannot express uncertainty/dispute in the graph

# Different layout algorithms

## Binary tree structure

**Pros:** commonly associated with genealogy, shows hierarchies and generations

**Cons:** Don’t deal with strange intergenerational connections. Certain that each entity will have more than one parent, and will possibly have more than 2 children. G tree is not valid either as does not deal with intergenerational.

## Cyclic graphs & Bidirectional Acyclic Graphs, Directed Acyclic Graphs,

Not possible. Person can’t be their own grandparent.

Needs to have both direct and undirected components. Don’t need to do operations on them yet so direction does not matter.

Maybe allow specification of directed or undirected connection? Hybrid?

## Force-directed layout for DAG

**Pros:**

Standard visualisation format for DAGs. Readily available for use and experimentation in different visualisation software, esp. D3.

**Cons:**

Nothing like genealogy graphs. Very few of them have formatting for genealogies. Need the hierarchical structure since otherwise it is unintuitive and doesn’t look like a genealogy chart (connections not immediately obvious)

Tried this, worked well as an early prototype but was clear that this was unintuitive. So started from scratch with a bespoke algorithm.

## Hierarchical layout for DAG

**Pros:**

Significantly closer to expectations for genealogy charts.

**Cons:**

Generally does not have support for more unusual relationships (e.g. intergenerational).

## Ancestor Fan

**Pros:**

Another well-known version of genealogy charts.

Could be interactable.

Immediately obvious who is the center entity / point of focus.

**Cons:**

Would not be able to show intergenerational incest, and is difficult to read.

Would be hard to modify

# Different relationship types considered

Insert the screenshots of the digitized plans for showing relationships.