

Travelling Salesperson Problem Project Plan

1810ICT Software Development Processes



Scott McMullan

s2916255

scott.mcmullan@griffithuni.edu.au

Contents

Project Plan	2
Introduction	2
Project Scope	2
Work Breakdown Structure	4
Activity Definition & Estimation.....	6
Gantt Chart	9

Project Plan

Introduction

The Travelling Salesperson Problem (TSP) is a classic problem in the field of computer science. Created by mathematicians William Rowan Hamilton and Thomas Kirkman in the 1800's, it was not until the 1930's that the problem was properly defined. Simply put, given several nodes (cities) and their locations, the problem is to find the shortest distance a salesperson would have to travel to visit each node. This is also referred to as the "cheapest" distance.

As it is a problem in optimization, it has been thoroughly studied and many solutions have been produced by different people around the world. Therefore, in this project there won't be a new solution developed, but existing algorithms will be implemented to solve it.

This document will outline the project scope, processes, a work breakdown structure, Gantt chart as well as a final report on the overall success of the project and the Python algorithm itself.

Project Scope

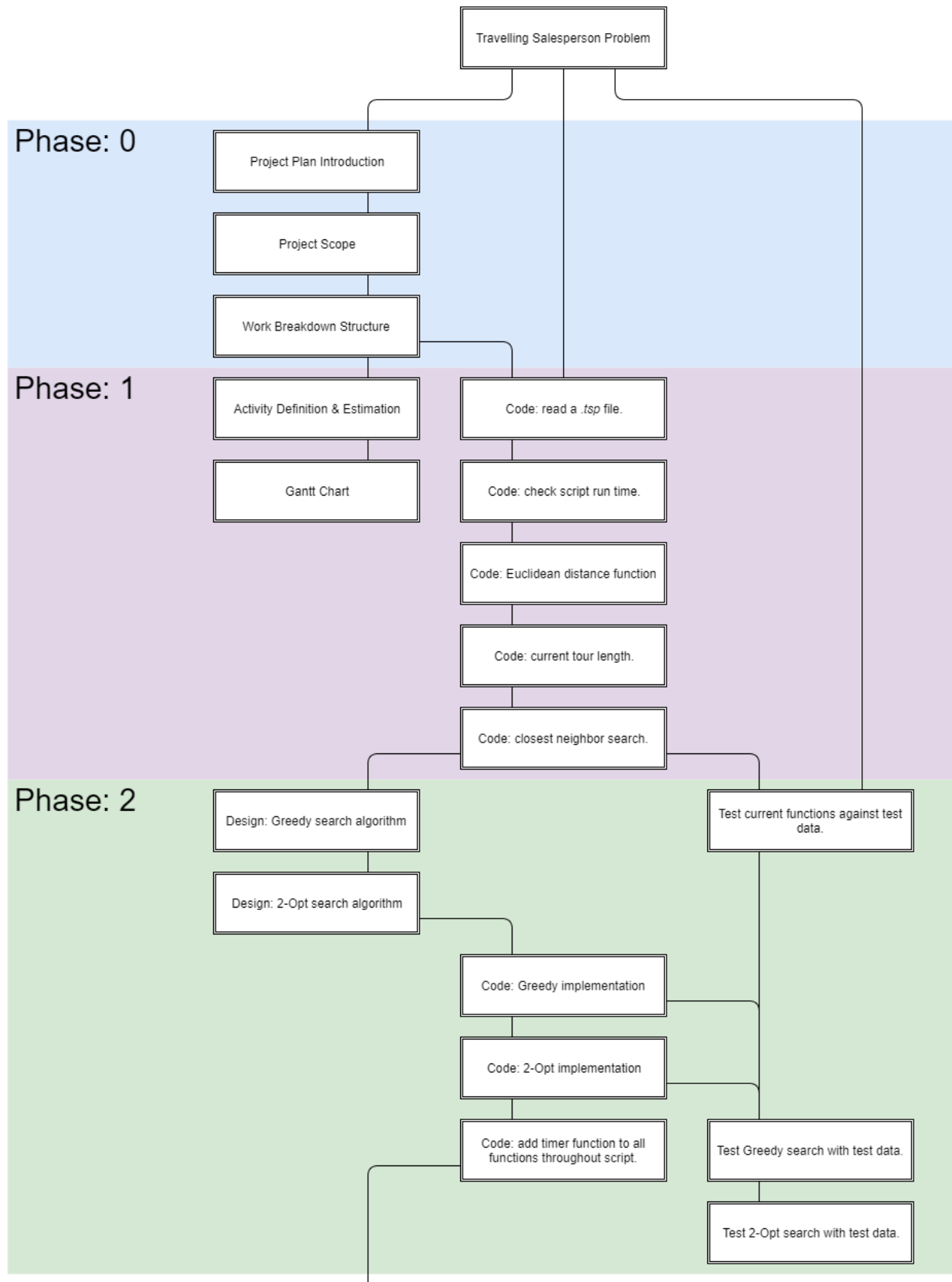
The objective of this project is to implement existing algorithmic solutions to the classic Travelling Salesperson Problem, using the Python programming language. A concise and efficient script needs to produce that satisfactorily solves the T.S.P. in numerous different cases with varying node amounts and distances. The script needs to be able to accept a time limit, in seconds, so that different solutions can be demonstrated based on the amount of time allowed. A database will also be a part of the T.S.P. solution; problems will be stored in one table, read from it, solved, and then the result will be stored in another table from which they can be retrieved. Along with all of the algorithmic solving of the T.S.P., database transactions will be handled by the Python script too.

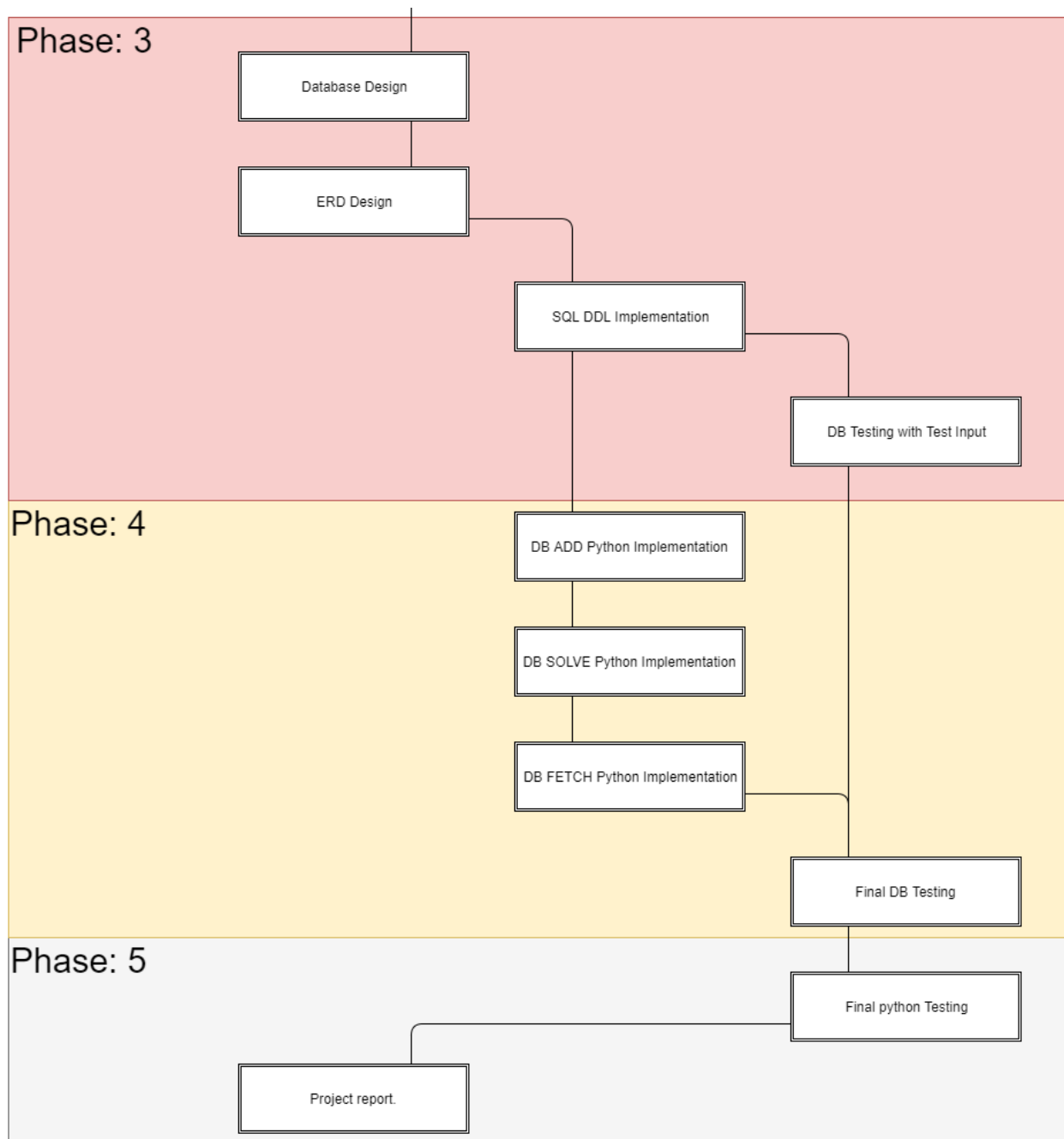
Development of the Python script and accompanying project document will be broken down into several phases:

0. This is where the project planning will happen. The outline for the Python script and database design will start. Along with those things, the work breakdown structure will be created and the project scope will be assessed.
1. Once the work breakdown structure has been satisfactorily completed, it will be used to design the project's Gantt chart and activity definition & estimation. While these two components are developed, work can start on the Python script;
 - The script will be written in small segments, built from the ground up, piece by piece. The first piece to be added will be reading a .tsp file and storing the necessary information.
 - Next, the script will add the ability to be timed during runtime.
 - A small function to calculate the Euclidean distance of two nodes will be added.
 - Another function to find the tour length of the current route of nodes will also be developed.
 - Lastly, a search function to find the closest neighbor of a node.
2. With the conclusion of the implementation of those basic script features, initial testing can begin on them. Also, in phase 2 the design and application of the two main algorithm can start;
 - Firstly, the pen & paper design for Greedy Search.
 - Secondly, pen & paper design for the 2-Opt Search.

- Once those designs are clear enough to be written into the script, the code will be updated with them. First Greedy and then 2-Opt will be put into the script.
 - After the addition of the two search algorithms has taken place, then can be tested on using some test data contrived for this purpose.
 - If testing is successful on the two algorithms, then a timer will be place in all functions that use them so that the entire script can be timed as needed.
3. After the python script is completed and up to standard, a database will be designed and integrated to work with the code.
 - Firstly, a pen & paper diagram of the database will be drawn up.
 - After this, a clear and concise entity relationship diagram will be developed.
 - When the database design is complete, the data definition will be coded using SQLite and the database itself will be created.
 - Finally, the database relations will be tested with test data.
 4. Each of the main functions needed to interact with the database will be added to the python script and the script itself will be modified where necessary. When this is complete, final database-script interaction testing will be performed.
 5. The final phase includes overall project testing and a report written on the project success and outcome.

Work Breakdown Structure





Activity Definition & Estimation

1. Project Management Plan	
Description: detailed and vital documentation to track project requirements, functionalities, scheduling and the phases and tasks involved with the project.	Time estimation: This activity will take 5 days to complete.
1.1. Introduction	
Description: Background information on the problem, including an outline and short history of the TSP.	Time estimation: This activity will take 2 days to complete.
1.2 Scope	
Description: Determining and documenting a list of specific project goals, deliverables, tasks, costs, and deadlines.	Time estimation: This activity will take 5 days to complete.
1.2.1 Project Requirements	
Description: Outlining all the requirements for the project; what are the deliverables and what needs to be achieved.	Time estimation: This activity will take 2 hours to complete.
1.2.2 Phase Definitions	
Description: A list of all the different phases in the project and a short description for each.	Time estimation: This activity will take 1 day to complete.
2. Work Breakdown Structure	
Description: An easy to read flow it items showing the separate entities of work and how they connect and continues from each other.	Time estimation: This activity will take 2 days to complete.
3. Gantt Chart	
Description: A graphically representation of the Work Breakdown Structure that shows more clearly the flow of work and the duration of each section.	Time estimation: This activity will take 2 days to complete.
4. (Part A): Code Implementation	
Description: The writing of the script to solve the Travelling Salesperson Problem.	Time estimation: This activity will take 17 days to complete.

4.1 File I/O

Description: Adding the ability to read in information from a file and organise it correctly.

Time estimation: This activity will take 2 days to complete.

4.2 Basic Arithmetic Functions

Description: Functions to calculate Euclidean distance, total tour length and to get the current process run-time.

Time estimation: This activity will take 5 days to complete.

4.3 Initial Testing

Description: Testing the code that has been implemented so far and making sure it runs properly.

Time estimation: This activity will take 1 day to complete.

4.4 Greedy Search Design

Description: Using pen and paper to design an implementation of Greedy search for the Python script. Also, development of pseudocode.

Time estimation: This activity will take 4 days to complete.

4.5 Greedy Search Implementation

Description: Taking the previous design from pen and paper and pseudocode and coding it into the Python script.

Time estimation: This activity will take 1 day to complete.

4.6 2-Opt Search Design

Description: Using pen and paper to design an implementation of 2-Opt search for the Python script. Also, development of pseudocode.

Time estimation: This activity will take 4 days to complete.

4.7 2-Opt Search Implementation

Description: Taking the previous design from pen and paper and pseudocode and coding it into the Python script.

Time estimation: This activity will take 1 day to complete.

4.8 Implement Timer

Description: Making sure that the algorithm is being timed as it executes.

Time estimation: This activity will take 1 day to complete.

4.9 (Part A): Final Testing

Description: Testing *all* aspects of the Python script, making sure everything is working and efficient.

Time estimation: This activity will take 2 days to complete.

5. Entity Relationship Diagram

Description: Create an ER diagram to base the design of the script database.

Time estimation: This activity will take 1 day to complete.

6. Database queries & Scripts

Description: Write all the possible queries that the Python script will use to interact with the database and get information.

Time estimation: This activity will take 2 days to complete.

8. Database Implementation

Description: Implement the physical database based on the design in the ER diagram created previously.

Time estimation: This activity will take 4 days to complete.

9. GUI Implementation

Description: Implement a GUI so that the interaction and output of the Python program can be shown via a graphical medium.

Time estimation: N/A

10. Final Project Testing

Description: Do a final test of the entire program with all components integrated and working.

Time estimation: This activity will take 1 day to complete.

11. Final Project Report

Description: Write the final report detailing the performance and outcome of the project.

Time estimation: This activity will take 3 days to complete.

Gantt Chart

Travelling Salesperson Problem: Gantt Chart

Select a period to highlight at right. A legend describing the charting follows.

Period Highlight: 1

Plan Duration

Actual Start

% Complete

Actual (beyond plan)

% Complete (beyond plan)

