## COMP 3600/6466 Final Project Report

#### Milestone - 1: Project Proposal

##### 综述:

本项目将实现 Which Path to Go? 程序, 它可以帮助人们找到两点之间的最短可通行路径, 算法将根据1.每条路径上所花费的时间, 2.该路径上是否有房间(结点)达到了最大容量, 导致无法通过. 算法的详细实现方案请看Assumption. 此外它还可以帮助人们在该图上建立费用最小的生成树, 可以为人们提供建立道路连通所有节点的解决方案.

**Assumptions:**

* 房间/地点(节点)之间由路径(边)连通, 其中每条路径具有通行开销(例如通行时间, 通行花费).
* 房间/地点达到最大容量后, 所有人不能通过该结点.
* 每个人在出发时有一定的体力值(允许的最大开销), 如果所经过的路径花费的总开销大于此人出发时的开销, 该路径将被标记为非法(不允许这样通行).
* 在提供有效路径之前执行查询的人员和执行路径完成查询的人员时, 区域中的人员数量不会改变. 但在执行点到点的路径后, 人员数量是动态更新的.
* 默认计算开销总不超过int32范围, 且每次查询的最优解总是存在的.
* 输入数据内容/格式包含: 房间/地点的位置信息, 即该房间与那些房间相连, 图结构信息; 每条路径(边, 点到点之间)的通行开销(例如通行时间, 通行花费); 房间/地点(节点)所能容纳的最大人数; 地图上的所有人员及他们的初始位置;
* 可执行的查询指令: 在某时刻人员A想要到达地点I的最优路径; 生成全图开销最小的生成树.

**Functionalities:**

* (标记, 将被计算分数的功能1): 该功能覆盖了Topic B.9.: MST+Dijkstra.
* 该功能提供生成整个地图关于边开销最小的生成树, 为连通图的最小道路建设开销提供方案. 时间复杂度为 , 其中是边集大小, 是节点集大小.
* (标记, 将被计算分数的功能2): 该功能覆盖了Topic B.8.: Dynamic Programming.
* 该功能基于图上的生成树进行动态规划, 查找根节点到目的节点的最短路径, 充分利用树结构的无后效性, 树上DP. 时间复杂度为
* (标记, 将被计算分数的功能3): 该功能覆盖了Topic B.4: Heap.
* 该功能能快速找到前 小的点到点通行路径, 为人们提供更多选择. 时间复杂度为 .
* I/O: 输入数据与查询, 程序建立数据结构并执行算法, 输出当前地图信息与查询结果. 查询可以动态循环多次.
* 动态更新图的各种信息, 包括节点人数, 通行方案等.