Algorithms and Data Structures 1

Daniel Brasholt s214675

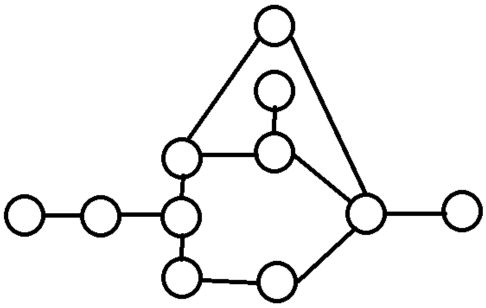
08.04.22

Hand-in Exercise: Floor Plan

Exercise 1.1

The floor plan can be modelled as a graph by building each room as a vertex in the graph and each door as an edge between the vertices that the door connects. One can then move from one room to another if and only if the two vertices are connected with an edge. This graph would then not be directional, since one can freely walk through a door in each direction.

Exercise 1.2



Exercise 1.3

If and only if there exists a path between the fire escape and each room in the house, the graph representing the floor plan is a connected graph. Using a depth-first search, starting from e, each vertex (room) should then be visited at least once. If this is not the case, the graph is not connected and an escape route cannot be found from each room.

Since some edges - the fire doors - are to be ignored, these must be factored into the search. In a DFS, each room is only visited once. For each room, all doors out of the room must be explored - only, however, if they are not fire doors. For each door, it must then be checked if the door is in the set of fire doors, which, in the worst case, takes $O(k)$ time. Since this must be done for each door, the total time complexity of the search is $O(R + kD)$.

Exercise 1.4

To produce a minimum cost set of doors such that all rooms are connected, an algorithm for finding a minimum spanning tree can be used. Kruskal's algorithm would produce and output the set of doors necessary to cover the entire floor while keeping the cost as low as possible. This algorithm has a complexity of $O(m \log n)$, where m is the number of edges (number of doors, D), and n the number of vertices (number of rooms, R). Therefore, the set of doors can be computed with a running time of $O(D \log R)$.