Algorithms and Data Structures 1

Daniel Brasholt s214675

18.03.22

Hand-in Exercise: Super Mario Run

Exercise 1.1

In a $k \times k$ grid, a total of k^2 fields exist. A field is, however, only valid if there is brick on the field underneath. This means that at most $\frac{1}{2}k^2$ fields can be valid. Using asymptotic notation, this is then $O(k^2)$ since the leading coefficient is ignored.

Each edge is defined as a valid move. Each node can have at most 2 valid moves; if the *forward* move is valid, there must be a brick at field $(x+1, y-1)$. Since this field has to be empty for Mario to fall, these two actions are mutually exclusive. Once again, the valid number of edges is then $O(k^2)$.

Exercise 1.2

Since all valid fields are in the graph, the goal must also be in the graph if it is possible to reach from the start. Therefore, the graph must be searched. If the goal field is at any point encountered during the search, it must be possible to reach. This can be done with a Depth-First Search, which has the time complexity of $O(n+m)$ where n is the number of nodes and m the number of edges. Using the information found in exercise 1.1, the running time of the search for the goal field can be written as

$$O(n+m) = O(k^2 + k^2) = O(2k^2) = O(k^2)$$

Exercise 1.3

Since the mushroom field must be visited before the end goal, finding the shortest path from start to goal that goes through c is equivalent to finding the shortest path between the start and c, and then the shortest path from c to the end goal. The shortest path can be found with the principles used in conducting a Breadth-First Search. For each node, all the node's edges are put into a queue. The first element in the queue is then visited. If the algorithm at any point encounters c, the rest of the queue is cleared and another BFS is conducted with c as the new starting node, searching for the end field. This algorithm has the running time of

$$O(2 \cdot (n+m)) = O(n+m) = O(k^2 + k^2) = O(k^2)$$