Algorithms and Data Structures 1

Student number: s215158
Hand-in: Exam simulation

1 Exercise

1.1

In order to model the ski resort as a graph we can simply imagine that every *position* is a node and the edges, represented by the *ski slopes*, are represented as directed edges from node *start position* to the node *end position* with the corresponding *completion time*.

1.2

To find if it's possible to get at the bottom of the mountain before the resort closes we can use Dijkstra's algorithm, starting from p_0, on the graph described in the previous section. At the end we will only need to check that the distance from p_0 to p_{x-1} is at most t, in that case it should return "True" and "False" otherwise. Dijkstra's algorithm guarantees that the distance of each node is the shortest path distance from the source node. If the distance of $p_{x-1} \leq t$, it means there exists a path from with a completion time at most t. The running time, with a minimum heap of the algorithm is: $O(y \cdot log(x))$ where y is the number of *sky slopes* and x the number of *positions*.

1.3

To find the shortest path with the lowest level possible we can again use Dijkstra's algorithm with a small modification. Every time, before adding an edge, we need to check if the level of the ski slope is less than or equal to ℓ, then we can update the distance. Otherwise, repeat the algorithm with the next level of difficulty $\ell + 1$ until you find the smallest level of difficulty that also satisfies the distance $p_0 \rightarrow p_{x-1} \leq t$. The algorithm ensures that it will find the right result since it only considers *ski slopes* of increasing difficulty until it finds the smallest one that satisfies the time limit t. The time complexity remains the same: $O(y \cdot log(x))$, we should however know that it might partially run ℓ times before finding the right result.