Blitz Bomb

The company JAV a.s is in the process of developing a video game known as Blitz Bomb. This game places you in the role of Kelvin, an alien hailing from the distant planet HOT 069. Kelvin is an unusual flame with a passionate desire to cause explosions. However, yesterday, Kelvin landed on Earth, specifically in the city of Cali, with a truly unique purpose: to trigger the detonation of a series of hidden bombs that are meticulously interconnected through a mysterious network of gunpowder placed by the extremist incendiary group to which Kelvin belongs. However, it is important to note that some of these bombs have already been deactivated by Cali's bomb squad.

In his dangerous mission, Kelvin must efficiently traverse the entire network of gunpowder pathways, activating all the bombs and reaching the exit before the police catch him. If Kelvin fails to trigger all the bombs and escape within the established time, the game will come to an end, and the authorities will prevail.

To accomplish this, a simple graph with a minimum of 50 vertices is implemented, where some vertices contain a bomb, and the gunpowder pathways are the edges. The graph must be connected and have no isolated vertices. The explosive graph will be generated by the program, which will also calculate the minimum time in which all vertices of the graph can be traversed optimally and reach the exit, thereby obtaining a time. The player must traverse the same graph within the maximum established time; otherwise, they will lose.

For this game, there will be three difficulty levels (easy, medium, hard). In the hard difficulty, the time Kelvin needs to traverse the graph will be calculated by the program (the most efficient time). In the medium difficulty, the player will have an additional 10 seconds to traverse the graph compared to the hard level, and in the easy difficulty, the player will have an additional 15 seconds compared to the hard level to traverse the graph.

The game will be capable of functioning with at least two different graph implementations. One version will use an adjacency matrix data structure, and the other will use an adjacency list data structure. Both versions must be interchangeable without affecting the game's functionality. Additionally, three prominent graph algorithms will be used: DFS, Dijkstra, and Prim.

The Prim and DFS algorithms will be used to calculate an approximate time that the user will take to traverse the entire graph. Prim will find the most optimal path, and DFS will sum all the weights, which represent the time from one node to another.

Finally, Dijkstra will be used as a power-up that Kelvin can use to determine the fastest route between two nodes. This power-up will be effective because the player has the possibility of reaching the exit without triggering all the bombs but will receive a penalty of 30 seconds for each unexploded bomb.