Problem 1 - Graphics

(a) Grid King and Albert

1. My algorithm is as follows:

   (a) Construct a dual-layer weighted graph: The gragh is as follows. Divide the grid into two layers. The first layer represents vertical movements, and the second layer represents horizontal movements. Edges within layers have a weight of a, and edges between layers have a weight of b.

   (b) Mark landmines: Set nodes with landmines to be unreachable, with an infinite weight for edges to any other nodes.

   (c) Execute the Dijkstra algorithm: Perform the Dijkstra algorithm from two starting points $(1, 1, 0)$ and $(1, 1, 1)$ for both layers to calculate the shortest time to each node.

   (d) Determine the final time: Compare the shortest times to reach the end nodes $(m, n, 0)$ and $(m, n, 1)$ and take the minimum of the two.

   (e) Output the result: Output the shortest time from the starting point $(1, 1)$ to the endpoint (m, n).

   Proof of Correctness: Based on the property of Dijkstra's algorithm to find the shortest path, our dual-layer weighted graph model accounts for all scenarios involving turns and straight movements. In this model, horizontal and vertical movements are assigned to separate layers, and turns are represented by edges connecting the two layers. Besides, we mark landmines with an infinite weight for edges to any other nodes, so we won't reach to landmines in the process of performing the Dijkstra algorithm. Since Dijkstra's algorithm guarantees the shortest path within a given weighted graph, our dual-layer graph model allows the algorithm to consider the cost of each turn. Thus, this algorithm correctly computes the shortest time from the starting point to the endpoint.

   Time Complexity: The time complexity for constructing the dual-layer graph is $O(mn)$, marking landmines is $O(c)$, and executing the Dijkstra algorithm is $O(mn\log(mn))$. The dominating factor is the Dijkstra algorithm, making the total complexity $O(mn(\log m + \log n))$.

2. My algorithm is as follows:

   (a) If d is greater than $m+n$, we can simply remove all landmines on the ideal shortest path.

   (b) If d is less than or equal to $m + n$, then the algorithm is as follows:

i. **Variable Definition:**
   - Initialize a four-dimensional array $\text{dis}[m][n][2][d]$, where $\text{dis}[w][x][y][z]$ represents the shortest distance to coordinate (w, x, y) with a potential to remove z more landmines.

ii. **Array Initialization:**
   - Set the origin to 0 and other points' dis values to infinity.

iii. **Running Dijkstra's Algorithm:**
   - For relaxing edge (u, v), consider two cases:
     A. If v is a mine and not a turn (i.e., weight $\neq b$):
     $$\text{dis}[w_v][x_v][y_v][i] = \min(\text{dis}[w_v][x_v][y_v][i], \text{dis}[w_u][x_u][y_u][i{+}1]{+}\text{weight of edge}(u, v))$$
     where $0 \leq i \leq d - 1$.
     B. In all other cases:
     $$\text{dis}[w_v][x_v][y_v][i] = \min(\text{dis}[w_v][x_v][y_v][i], \text{dis}[w_u][x_u][y_u][i]{+}\text{weight of edge}(u, v))$$
     where $0 \leq i \leq d$.

iv. **Result:**
   - The answer is $\min\{\text{dis}[m][n][0][i], \text{dis}[m][n][1][i]\}, 0 \leq i \leq d$.

Proof of Correctness: Since in the previous question, we established that the algorithm can find the shortest distance in a field with d landmines, our current algorithm records different quotas, thus considering all scenarios. This means choosing the best case among all possible mine removal methods. Furthermore, it's impossible to remove more than d landmines, confirming the algorithm's correctness.

Time Complexity: When d is greater than $m{+}n$, we can directly obtain the answer, making the complexity $O(1)$. In other cases, we perform the Dijkstra Algorithm at most d times on the entire graph. Given the maximum value of d is $m + n$, we perform the Dijkstra Algorithm at most $m + n$ times. Therefore, the complexity is $O(mn(m + n)(\log(m) + \log(n)))$.

The following text shows how to write the math formula:

Theorem (Leibniz integral rule)**.** Let $f(x, t)$ and its partial derivative $\frac{\partial}{\partial x} f(x, t)$ be continuous in x and t in some region of the (x, t) plane that includes $a(x) \leq t \leq b(x)$ and $x_0 \leq x \leq x_1$. Suppose also that the functions $a(x)$ and $b(x)$ are continuous and have continuous derivatives for $x_0 \leq x \leq x_1$. Then, for $x_0 \leq x \leq x_1$, we have:
$$\frac{\mathrm{d}}{\mathrm{d}x} \int_{a(x)}^{b(x)} f(x, t)\,\mathrm{d}t = f(x, b(x)) \cdot \frac{\mathrm{d}}{\mathrm{d}x} b(x) - f(x, a(x)) \cdot \frac{\mathrm{d}}{\mathrm{d}x} a(x) + \int_{a(x)}^{b(x)} \frac{\partial}{\partial x} f(x, t)\,\mathrm{d}t$$
If $a(x) = a$ and $b(x) = b$, where a and b are constants, then the above reduces to:
$$\frac{\mathrm{d}}{\mathrm{d}x} \int_a^b f(x, t)\,\mathrm{d}t = \int_a^b \frac{\partial}{\partial x} f(x, t)\,\mathrm{d}t$$

References

[1] Last name, first name. *Book Title*. Publisher, Year. Print.

[2] Last name, first name. “Webpage Title”. Website name, Organization name. Online; accessed Month Date, Year. `www.URLhere.com`