<u>Problem 1:</u>

Imagine the BFS progressing as frontiers. You take a starting vertex `s`, which is at level `0`. All the adjacent vertices are at level `1`. Then, we mark all the adjacent vertices of all vertices at level 1, which don't have a level, to level `2`. So, every vertex will belong to one frontier (or level) only. And when an element is in a frontier, we check once for its adjacent vertices, which takes O(|V|) time. As, the frontier covers |V| elements over the course of the algorithm, the total time would become O(|V| * |V|) which is $O(|V|^2)$.

Yes Complexity is reduced when adjacency list is used O(V+E)

Proof :

`v1 + (incident edges) + v2 + (incident edges) + .... + vn + (incident edges)`

Can be rewritten as:

`(v1 + v2 + ... + vn) + [(incident_edges v1) + (incident_edges v2) + ... + (incident_edges vn)]`

Thus O(V+E)

Rubric :

Proof of correctness for BFS using adjacency matrix : 2 marks

Comparison with adjacency list and complete derivation of time complexities of both : 3 marks. Marks not alloted for just stating the time complexity and not proving.

<u>Problem-2</u>

The idea is to consider the snake and ladder board as directed graph and run BFS from starting node which is vertex 0 as per game rules. We construct a directed graph keeping in mind below conditions

1. For any vertex in the graph v, we have an edge from v to v + 1, v + 2, v + 3, v + 4, v + 5, v + 6 as we can reach any of these nodes in one throw of dice from node v.

2. If any of these neighbors of v has a ladder or snake which takes you to position x, then x becomes the neighbor instead of the base of the ladder or head of the snake.

<u>Rubric:</u>

Idea and algorithm: 7 marks

Proof of correctness:5 marks

Time complexity and justification for best algorithm:5 marks

<u>Problem-3</u>

Standard flood fill algorithm.(https://en.wikipedia.org/wiki/Flood_fill)

Rubric :

Algorithm and Pseudocode : 8 marks (subjective)

Correctness : 8 marks (subjective)
Complexity : 2 marks