Mario and the Mysterious Bridge

•

To tackle this problem you need to have some knowledge on simple graph theory concepts like vertices and edges . Here we need to map the problem into a directed graph . First you need to create the graph and the road which bridge runs. To create the bridge you can use a vector or an array of size n according to the given end index. When there is a blue brick as in the problem statement you are going to a lower index that means there must be an edge starting from that block to [block – number on the block] .
. And when there is a red block there must be an edge starting from the particular red block to block + number in the block.
As I mentioned above we need to map all the indexes to a directed graph accordingly skipping broken blocks . And the number of maximum distance he can jump must be considered when building bridge.

```
int min_jumps(int n, vector<pair<int, int> > red, vector<pair<int, int> > blue, vector<int> broken, int maxJmp)
{
    vector<int> board(n + 1, 0);
    vector<int> brokenPlaces(n + 1, 0);
    for (int i = 0; i < broken.size(); i++)
    {
        brokenPlaces[broken[i]] = 1;
    }
    // board to graph conversion
    for (auto sp : blue)
    {

        int s = sp.first;
        int e = sp.second;
        if (!(brokenPlaces[s]) && !(brokenPlaces[e-s]))
        // if (!(brokenPlaces[s]))
            board[s] = -1 * e;
    }

    for (auto lp : red)
    {
        int s = lp.first;
        int e = lp.second;
        if (!(brokenPlaces[s]) && !(brokenPlaces[e+s]))
        // if (!(brokenPlaces[s]))
            board[s] = e;
    }

    // Graph
```

```
    Graph g(n + 1);
    for (int u = 1; u < n; u++)
    {
        for (int jmp = 1; jmp <= maxJmp; jmp++)
        {
            int v = u + jmp;
            v += board[v];
            if (v <= n && (!brokenPlaces[v]))
            {
                g.addEdge(u, v);
            }
        }
    }
    return g.minCostBFS(1, n);
}
```

And once you have mapped the red blocks and blue blocks considering broken blocks and maximum jumps to a graph you just have to call a shortest path algorithm like bfs or dijkstra's algorithm .