Assignment 1

[25 Marks]

Deadline: August 27, 2023

1 Problems on sets- union, member, intersection and power set. [5 Marks]

1.1 Given two integer lists nums1 and nums2, return a list of their intersection. Each element in the result must be unique and you may return the result in any order

1.1.1 Examples:

- **Input**: nums1 = [4, 9, 5], nums2 = [9, 4, 9, 8, 4]
- **Output**: [9, 4]

1.2 Given two lists nums1 and nums2,return a list of their union. Union of the two lists can be defined as the set containing distinct elements from both the lists. If there are repetitions, then only one occurrence of the element should be printed in the union.

1.2.1 Examples:

- **Input**: nums1 = [4, 9, 5], nums2 = [9, 4, 9, 8, 4]
- **Output**: [4, 9, 5, 8]

1.3 Given an integer list nums of unique elements, return all possible subsets(the power set). The solution set must not contain duplicate subsets. Return the solution in any order.

1.3.1 Examples:

- **Input**: nums = [1, 2, 3]
- **Output**: [[], [1], [2], [1, 2], [3], [1, 3], [2, 3], [1, 2, 3]]

2 Game of SplitList [10 Marks]

Given a List L of size k and a value x (an integer). Can you determine if it is possible to split the list into k non-empty lists? At each step of the split, one of the following condition has to be satisfied for each *subList*:

- The length of the *subList* is one, or
- The sum of elements of the *subList* is greater than or equal to x.

Your implementation should return one if it is possible to split the input list, otherwise return zero.

Note: A subList is a contiguous non-empty sequence of elements within a List.

2.1 Sample Input and Output

1. Example
   - **Input**: L = [2, 2, 1], x = 4
   - **Output**: true
   - **Explanation**: We can split the array into [2, 2] and [1] in the first step. Then, in the second step, we can split [2, 2] into [2] and [2]. As a result, the answer is true.
2. Example
   - **Input**: L = [2, 1, 3], x = 5
   - **Output**: false
   - **Explanation**: We can try splitting the array in two different ways: the first way is to have [2, 1] and [3], and the second way is to have [2] and [1, 3]. However, both of these ways are not valid. So, the answer is false.

3 Save your car from boulders [10 Marks]

You are provided with a 2D matrix grid of size $n \times n$, where each cell (r, c) holds either a boulder (if $grid[r][c] = 1$) or is smooth (if $grid[r][c] = 0$).

You begin at the starting cell $(0, 0)$. With each move, you can transition to any adjacent cell in the grid, including those occupied by the boulder.

The safety measure of a path on the grid is defined as the shortest Manhattan distance from any cell in the path to any boulder in the grid.

Determine the highest attainable safety measure among all the paths leading to the final cell $(n-1, n-1)$. A cell adjacent to the cell (r, c) is one of the following: $(r, c+1)$, $(r, c-1)$, $(r+1, c)$, or $(r-1, c)$, provided it exists. The diagonal movement is not allowed, for example, if you are at cell (r, c), the following moves are restricted: $(r-1, c-1)$, $(r+1, c+1)$, $(r-1, c+1)$ and $(r+1, c-1)$.

The Manhattan distance between two cells (a, b) and (x, y) is the sum of the absolute differences between their coordinates: $|a - x| + |b - y|$.

Constraints:

- $1 <= grid.length == n <= 5$.
- $grid[i].length == n$.
- $grid[i][j]$ is either 0 or 1.
- There is at least one boulder in the grid.

3.1 Sample Input and Output

1. Example
    - **Input**: grid = [[1, 0, 0], [0, 0, 0], [0, 0, 1]]
    - **Output**: 0
    - **Explanation**: All paths from (0, 0) to (n - 1, n - 1) go through the boulders in cells (0, 0) and (n - 1, n - 1).
2. Example
    - **Input**: grid = [[0, 0, 0, 1], [0, 0, 0, 0], [0, 0, 0, 0], [1, 0, 0, 0]]
    - **Output**: 2
    - **Explanation**: The path (0, 0) → (0, 2) → (1, 1) → (1, 2) → (2, 2) → (3, 2) → (3, 3), where (i, j) is the index of the cell in grid, has a safeness factor of 2 since:
      - The closest cell of the path to the boulder at cell $(0, 3)$ is cell $(1, 2)$. The distance between them is $|0 - 1| + |3 - 2| = 2$.
      - The closest cell of the path to the boulder at cell $(3, 0)$ is cell $(3, 2)$. The distance between them is $|3 - 3| + |0 - 2| = 2$.
      It can be shown that there are no other paths with a higher safeness factor.