Please write a Python script to solve the following problem. It should read the input from a file input.txt, which has the same format as the example:

You all arrive at a Lava Production Facility on a floating island in the sky. As the others begin to search the massive industrial complex, you feel a small nose boop your leg and look down to discover a reindeer wearing a hard hat.

The reindeer is holding a book titled "Lava Island Hiking Guide". However, when you open the book, you discover that most of it seems to have been scorched by lava! As you're about to ask how you can help, the reindeer brings you a blank topographic map of the surrounding area (your puzzle input) and looks up at you excitedly.

Perhaps you can help fill in the missing hiking trails?

The topographic map indicates the height at each position using a scale from 0 (lowest) to 9 (highest). For example:

```
0123
1234
8765
9876
```

Based on un-scorched scraps of the book, you determine that a good hiking trail is as long as possible and has an even, gradual, uphill slope. For all practical purposes, this means that a hiking trail is any path that starts at height 0, ends at height 9, and always increases by a height of exactly 1 at each step. Hiking trails never include diagonal steps - only up, down, left, or right (from the perspective of the map).

You look up from the map and notice that the reindeer has helpfully begun to construct a small pile of pencils, markers, rulers, compasses, stickers, and other equipment you might need to update the map with hiking trails.

A trailhead is any position that starts one or more hiking trails - here, these positions will always have height 0. Assembling more fragments of pages, you establish that a trailhead's score is the number of 9-height positions reachable from that trailhead via a hiking trail. In the above example, the single trailhead in the top left corner has a score of 1 because it can reach a single 9 (the one in the bottom left).

This trailhead has a score of 2:

```
...0...
...1...
...2...
6543456
```

7.....7
8.....8
9.....9
(The positions marked . are impassable tiles to simplify these examples; they do not appear on your actual topographic map.)

This trailhead has a score of 4 because every 9 is reachable via a hiking trail except the one immediately to the left of the trailhead:

..90..9
...1.98
...2..7
6543456
765.987
876....
987....
This topographic map contains two trailheads; the trailhead at the top has a score of 1, while the trailhead at the bottom has a score of 2:

10..9..
2...8..
3...7..
4567654
...8..3
...9..2
.....01
Here's a larger example:

89010123
78121874
87430965
96549874
45678903
32019012
01329801
10456732
This larger example has 9 trailheads. Considering the trailheads in reading order, they have scores of 5, 6, 5, 3, 1, 3, 5, 3, and 5. Adding these scores together, the sum of the scores of all trailheads is 36.

The reindeer gleefully carries over a protractor and adds it to the pile. What is the sum of the scores of all trailheads on your topographic map?

Here's a Python script to solve the problem. It reads the topographic map from `input.txt` and calculates the sum of the scores of all trailheads.

Hiking Trails

Explanation of the Script

4. **Input Parsing**: The `parse_input` function reads the `input.txt` file and converts the map into a 2D list of integers.
2. **Finding Trailheads**: The `find_trailheads` function locates all positions on the map with a height of `0`.
3. **Breadth-First Search (BFS)**: The `bfs` function starts at a trailhead and explores all possible paths that follow the rules, counting reachable `9`s.
4. **Total Score Calculation**: The `calculate_total_score` function iterates over all trailheads, using BFS to calculate each trailhead's score and summing the scores.
5. **Main Function**: The `main` function ties everything together and prints the total score.

Save your topographic map in a file named `input.txt` and run this script to get the desired result. If you need further assistance or adjustments, feel free to ask!

2/2

Please update the script to solve part 2 of the problem:

The reindeer spends a few minutes reviewing your hiking trail map before realizing something, disappearing for a few minutes, and finally returning with yet another slightly-charred piece of paper.

The paper describes a second way to measure a trailhead called its rating. A trailhead's rating is the number of distinct hiking trails which begin at that trailhead. For example:

.....0.
..4321.
..5..2.
..6543.
..7..4.
..8765.
..9....
The above map has a single trailhead; its rating is 3 because there are exactly three distinct hiking trails which begin at that position:

.....0.   .....0.   .....0.
..4321.   .....1.   .....1.
..5....   .....2.   .....2.
..6....   ..6543.   .....3.
..7....   ..7....   .....4.
..8....   ..8....   ..8765.
..9....   ..9....   ..9....
Here is a map containing a single trailhead with rating 13:

..90..9
...1.98
...2..7

```
6543456
765.987
876....
987....
```

This map contains a single trailhead with rating 227 (because there are 121 distinct hiking trails that lead to the 9 on the right edge and 106 that lead to the 9 on the bottom edge):

```
012345
123456
234567
345678
4.6789
56789.
```

Here's the larger example from before:

```
89010123
78121874
87430965
96549874
45678903
32019012
01329801
10456732
```

Considering its trailheads in reading order, they have ratings of 20, 24, 10, 4, 1, 4, 5, 8, and 5. The sum of all trailhead ratings in this larger example topographic map is 81.

You're not sure how, but the reindeer seems to have crafted some tiny flags out of toothpicks and bits of paper and is using them to mark trailheads on your topographic map. What is the sum of the ratings of all trailheads?

Hiking Trails

```python
def parse_input(file_name):
  """Reads the input file and parses the topographic map."""
  with open(file_name, 'r') as file:
    return [list(map(int, line.strip())) for line in file.readlines()]

def find_trailheads(map_data):
  """Find all trailheads (positions with height 0)."""
  trailheads = []
  for row in range(len(map_data)):
    for col in range(len(map_data[0])):
      if map_data[row][col] == 0:
        trailheads.append((row, col))
  return trailheads

```

Edited

The script now calculates both the score (number of reachable 9s) and the rating (number of distinct hiking trails) for each trailhead. It prints the total score and rating for all trailheads. You can run the updated script to compute the results for your topographic map. Let me know if you need further assistance!

Message ChatGPT