2d 23h left

BETA Can't read the text? Switch theme

ALL

1

1. ACSL 2248

PROBLEM STATEMENT:

The ACSL version of the popular 2248 numbers game uses an 8 x 5 board with tiles containing the powers of 2. Initially, the board contains a random set of tiles with values up to and including 256. In each round of the game, you need to find the longest valid path on the board; a valid path must obey the following rules:

1. The first 2 tiles in the path must have the same value.
2. Each new tile added to the path must have the same value as the previous tile or have a value of the next increasing power of 2.
3. Look for a new tile in all 8 directions, ensuring a tile can only be used once in a path.

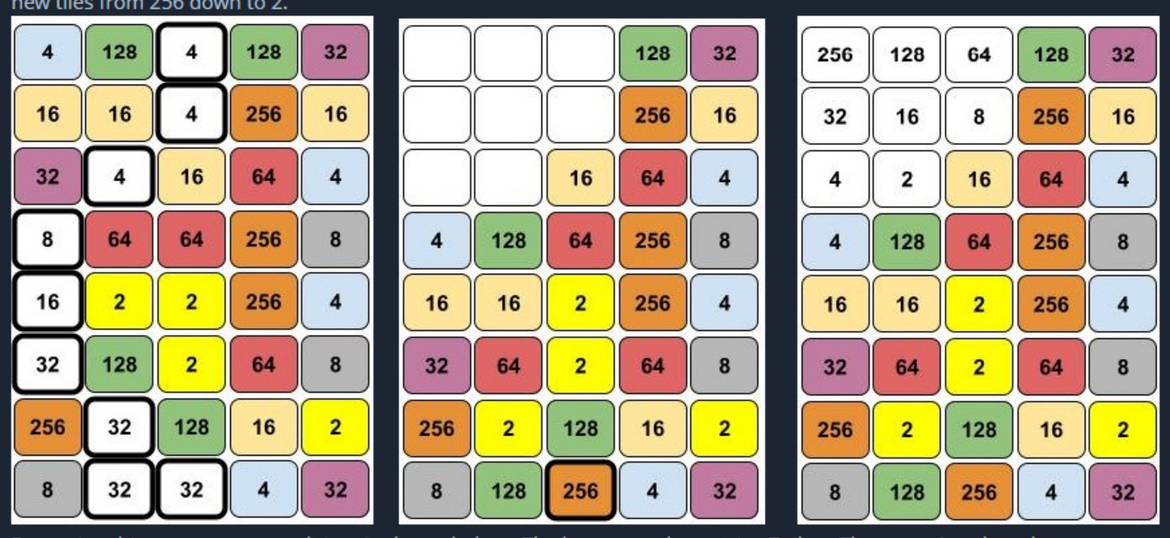
After you have found the longest path do the following:

1. All tiles in the path except the last one are removed from the board.
2. The final tile in the path is replaced by the power of 2 that is the smallest power of 2 that is greater than or equal to the sum of all of the tiles in the path.
3. Only 8 powers of 2 can exist on the board at any time. If the new tile added is larger than the largest power of 2 on the board already, then it becomes the largest tile that can be used to replenish the board. For example, if 1024 = 2^{10} is added to the board, all 2s and 4s are also removed from the board, since the only eight powers of 2 that can be on the board are 1024, 512, 256, 128, 64, 32, 16, and 8.
4. All of the tiles in a column above empty locations are moved down to fill those empty locations.
5. Finally, the empty locations at the top of the grid are filled in row-major order with new tiles with values using 8 decreasing powers of 2, cycling back to the largest number if necessary.

Play 3 rounds and then output the resulting board.
If there are multiple paths that tie for the longest, do the following: write the path as a string. where the location of each tile is written as a 2-digit number representing the row (1-8) and column (1-5). Use the path that sorts first alphabetically.

EXAMPLE:

In the board below, the longest path contains 9 tiles. There are two such paths; coincidentally, both use the same tiles (shown in white), but follow different routes:
13-23-32-41-51-61-72-82-83 and 13-23-32-41-51-61-72-83-82.
Sorting these paths, the first one ends up first, so it is used. The sum of the tiles in the path is 164. In the second image, the final tile in the path has been replaced with 256, the power of 2 that is greater than or equal to 164, and all other tiles in the path have been removed. The remaining tiles in each column shift downward. Finally, in the third image, all empty locations have been replaced with new tiles from 256 down to 2.



Repeating this process a second time is shown below. The longest path contains 7 tiles. There are 4 such paths:
51-52-61-62-73-82-83
51-52-61-62-73-82-71
52-51-61-62-73-82-83
52-51-61-62-73-82-71
All 4 paths include the same 6 tiles at the beginning, but differ in the 7th tile. When sorting the 4 paths, the underlined path is first, and it is the one chosen. The tiles sum to 640, so that last tile in the path is replaced with 1024. Remove the other tiles in the path, remove all 2s and 4s (because the 8 powers of 2 valid on the board are 1024, 512, ..., 8), and drop the other tiles. See the middle image. Replacing all empty blocks with powers of 2 from 1024 down to 8 yields the third board:

Repeating this process a final time is shown below. The longest path is 8-8-16-32-64-64-64-128-128-128-256-512-1024. The sum is 2432 so the next higher power of 2 is 4096 = 2^{12}. Therefore, the path and the blocks that are 8 or 16 are removed. They are replaced with numbers from 4096 down to 32. The boards below illustrate the different steps in the process:

The final board is output as:

```
4096 2048 1024 512 256 4096 128 64 32 4096 32 2048 1024 512 256 512 128 64 32 4096 32 2048 1024 256 512
256 256 256 256 64 32 128 64 256 1024 1024 128 256 64 32
```

TASK:

Complete the function **play2248**

- The function has 1 parameter: a string, *boardValues*, representing the 40 numbers on the original board, each separated by a single space.
- The function returns a string that represents the 40 numbers on the final board in row-major order, each separated by a single space.

You may create additional functions that are called from **play2248** if needed in solving the problem.

CONSTRAINTS:

Every number in each input string will be a power of 2 from 2 to 256.

DATA PROVIDED:

There are 5 sets of Sample Data for debugging and 5 sets of Test Data for scoring. You may create additional data sets for debugging your program.

Language Python 3 — Autocomplete Ready

Environment

```
1  #!/bin/python3 ...
10 #
11 # Complete the 'play2248' function below.
12 #
13 # The function is expected to return a STRING.
14 # The function accepts STRING boardValues as parameter.
15 #
16
17 def play2248(boardValues):
18     # Write your code here
19
20 if __name__ == '__main__':
21     fptr = open(os.environ['OUTPUT_PATH'], 'w')
22
23     boardValues = input()
24
25     result = play2248(boardValues)
26
27     fptr.write(result + '\n')
28
29     fptr.close()
30
```

Line: 20 Col: 27

Test Results | Custom Input | Run Code | Run Tests | Submit