Pacific atlantic

Got it! This is the Pacific Atlantic Water Flow problem. Let's break it down in very simple steps so you clearly understand what's happening.

✅ Problem in simple words

You're given a grid of heights.

- Pacific Ocean touches the top and left edges.
- Atlantic Ocean touches the bottom and right edges.

Water can flow from a cell to another neighbor cell if the neighbor's height is lower or equal (water flows downhill or stays level).

We need to find all cells where water can flow to BOTH oceans.

✅ Intuition

Imagine pouring water from the oceans into the grid:

- If you pour water from the Pacific, it can flow inland but only to equal or higher heights (reverse thinking).
- Same for the Atlantic.

Finally, take the intersection of cells reachable from both oceans.

✅ Step-by-Step Example

We have this height matrix:

csharp Copy Edit

```
[
  [1, 2, 2, 3, 5],
  [3, 2, 3, 4, 4],
  [2, 4, 5, 3, 1],
  [6, 7, 1, 4, 5],
  [5, 1, 1, 2, 4]
]
```

✅ Step 1: Start DFS from Pacific edges

Pacific is top row & left column.

- So we start DFS from:
  - Row 0 → `(0,0), (0,1), (0,2), (0,3), (0,4)`
  - Column 0 → `(0,0), (1,0), (2,0), (3,0), (4,0)`

From `(0,0)` (height 1):

- Can go to `(1,0)` (3) because 3 ⩾ 1
- From `(1,0)` → `(2,0)` (2) ❌ (2 < 3, cannot go uphill in reverse flow)
- From `(1,0)` → `(1,1)` (2) ❌

From `(0,1)` (2):

- Can go to `(0,2)` (2) → `(0,3)` (3) → `(0,4)` (5)
- And so on…

Eventually, DFS/BFS covers all cells that can be reached from Pacific.

Let's call this pacific_reachable set.

✅ Step 2: Start DFS from Atlantic edges

Atlantic is bottom row & right column.

- Start DFS from:
  - Row 4 → `(4,0), (4,1), (4,2), (4,3), (4,4)`
  - Column 4 → `(0,4), (1,4), (2,4), (3,4), (4,4)`

From `(4,4)` (4):

- Can go to `(3,4)` (5) → `(1,4)` (4) → `(0,4)` (5)
- Can also go left `(4,3)` (2) ❌ (lower height, can't reverse uphill)

From `(3,4)` (5):

- Can go to `(3,3)` (4) ❌
- Can go to `(2,2)` (5)

Eventually, this covers all atlantic_reachable set.

✅ Step 3: Intersection

Now, which cells are in BOTH pacific_reachable & atlantic_reachable?
Those are the cells where water can flow to both oceans.

Answer for this example:

```
[[0,4],[1,3],[1,4],[2,2],[3,0],[3,1],[4,0]]
```

These cells can reach both Pacific and Atlantic.

✅ Why this works?

Instead of checking every cell → can it flow to Pacific & Atlantic?
We reverse the flow:

- From the ocean, move inward only if next cell is higher/equal.
- This ensures we only visit valid cells where water can flow out.

So we do two DFS/BFS traversals and take the intersection.

✅ Complexity

- Time → O(M×N), since each cell is visited at most twice (Pacific & Atlantic).
- Space → O(M×N) for visited sets & recursion stack.

✅ Visual analogy

Imagine you're standing at the ocean and you're allowed to walk inland only to higher or equal ground.
Wherever you can walk inland from both oceans, that's the answer.

Would you like me to:

✅ Draw a visual diagram of Pacific vs Atlantic coverage?
✅ Or show you a step-by-step DFS code with comments?

Got it! Let's focus only on the INTERSECTION part in very simple terms.

✅ What is an intersection in this problem?

- After DFS/BFS from Pacific, we get a set of cells `pacific_reachable` → all cells that water can flow to the Pacific.
- After DFS/BFS from Atlantic, we get a set `atlantic_reachable` → all cells that water can flow to the Atlantic.

Now, some cells appear in BOTH sets.

Those cells are the intersection → they can reach both oceans.

✅ Small Example

Imagine a 3x3 grid:

Copy Edit

```
1 2 3
4 5 6
7 8 9
```

- Pacific touches top & left
- Atlantic touches bottom & right

1️⃣ Cells reachable from Pacific (top & left)

- Start from top row → `(0,0), (0,1), (0,2)`
- Start from left column → `(0,0), (1,0), (2,0)`

Water from Pacific can move to higher/equal heights, so it will reach:

`pacific_reachable = {(0,0), (0,1), (0,2), (1,0), (1,1), (1,2), (2,0)}`

2️⃣ Cells reachable from Atlantic (bottom & right)

- Start from bottom row → `(2,0), (2,1), (2,2)`
- Start from right column → `(0,2), (1,2), (2,2)`

Water from Atlantic can move to higher/equal heights, so it will reach:

`atlantic_reachable = {(2,2), (2,1), (2,0), (1,2), (1,1), (0,2)}`

3️⃣ INTERSECTION (reachable by BOTH)

Now take cells that are in BOTH sets:

```
pacific_reachable    atlantic_reachable =
 {(0,2), (1,1), (1,2), (2,0)}
```

✅ These are the cells where water can flow to BOTH oceans.

✅ How does code find the intersection?

It's just set intersection:

```python
result = pacific_reachable & atlantic_reachable
```

or in plain words:

Keep only the cells that are reachable from both oceans.

✅ Real-life analogy

- Imagine you can walk inland from Pacific and also walk inland from Atlantic.
- If a hill can be reached from BOTH ocean walks, it means rainwater on that hill can flow both ways.

Would you like me to:

✅ Show a tiny 2x2 or 3x3 visual example with full Pacific & Atlantic DFS sets drawn step by step?
✅ Or write small Python code showing how sets are built and intersect?

Which one would help you more?