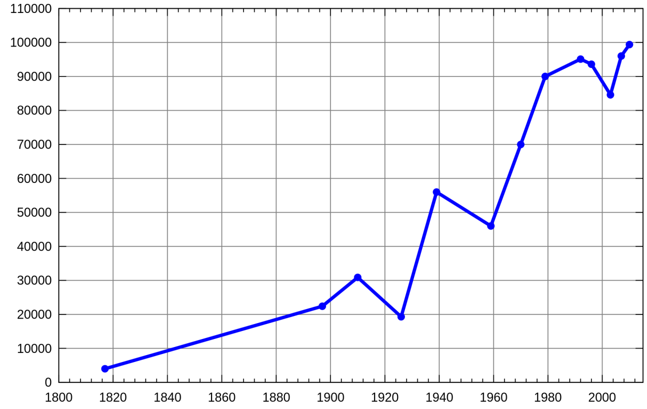
6076. Minimum Lines to Represent a Line Chart

My Submissions (/contest/weekly-contest-294/problems/minimum-lines-to-represent-a-line-chart/submissions/)

Back to Contest (/contest/weekly-contest-294/)

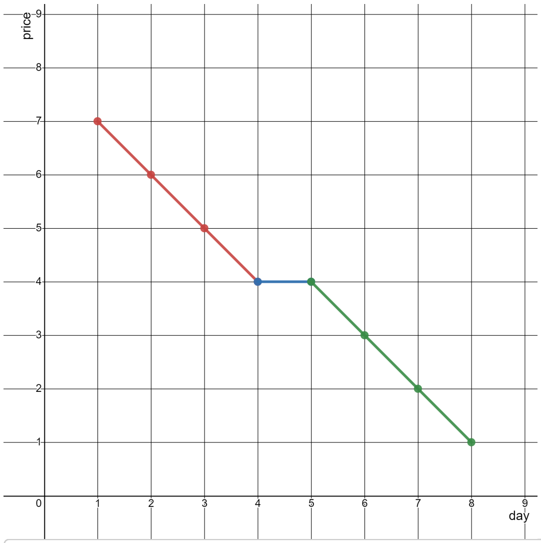
User Accepted:	79
User Tried:	180
Total Accepted:	79
Total Submissions:	245
Difficulty:	Medium

You are given a 2D integer array `stockPrices` where `stockPrices[i] = [day`$_i$`, price`$_i$`]` indicates the price of the stock on day `day`$_i$ is `price`$_i$. A **line chart** is created from the array by plotting the points on an XY plane with the X-axis representing the day and the Y-axis representing the price and connecting adjacent points. One such example is shown below:



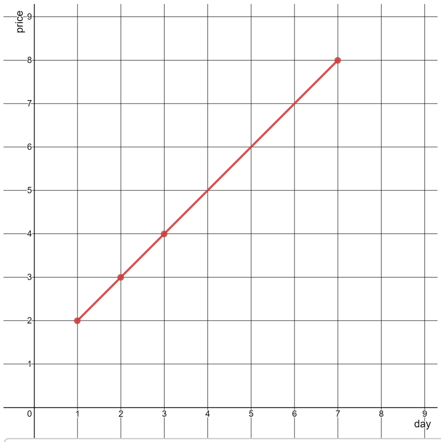
Return *the **minimum number of lines** needed to represent the line chart*.

Example 1:



```
Input: stockPrices = [[1,7],[2,6],[3,5],[4,4],[5,4],[6,3],[7,2],[8,1]]
Output: 3
Explanation:
The diagram above represents the input, with the X-axis representing the day and Y-axis representing the price.
The following 3 lines can be drawn to represent the line chart:
- Line 1 (in red) from (1,7) to (4,4) passing through (1,7), (2,6), (3,5), and (4,4).
- Line 2 (in blue) from (4,4) to (5,4).
- Line 3 (in green) from (5,4) to (8,1) passing through (5,4), (6,3), (7,2), and (8,1).
It can be shown that it is not possible to represent the line chart using less than 3 lines.
```

Example 2:



```
Input: stockPrices = [[3,4],[1,2],[7,8],[2,3]]
Output: 1
Explanation:
As shown in the diagram above, the line chart can be represented with a single line.
```

Constraints:

- `1 <= stockPrices.length <=` 10^5
- `stockPrices[i].length == 2`
- `1 <=` day_i, price_i `<=` 10^9
- All day_i are **distinct**.

JavaScript

```javascript
const ll = BigInt;
const minimumLines = (stockPrices) => {
    let n = stockPrices.length, res = 1;
    if (n == 1) return 0;
    stockPrices.sort((x, y) => x[0] - y[0]);
    for (let i = 1; i < n - 1; i++) {
        let [preX, preY] = stockPrices[i - 1], [curX, curY] = stockPrices[i], [nextX, nextY] = stockPrices[i + 1];
        if (ll((curX - preX)) * ll((nextY - curY)) != ll((nextX - curX)) * ll((curY - preY))) res++;
    }
    return res;
};
```

☐ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/704536206/) More Details (/submissions/detail/704536206/)

Share your acceptance!

4

Copyright © 2022 LeetCode

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)