

Problem 646: Maximum Length of Pair Chain

Problem Information

Difficulty: Medium

Acceptance Rate: 61.34%

Paid Only: No

Tags: Array, Dynamic Programming, Greedy, Sorting

Problem Description

You are given an array of n pairs `pairs` where `pairs[i] = [lefti, righti]` and `lefti < righti`.

A pair `p2 = [c, d]` **follows** a pair `p1 = [a, b]` if `b < c`. A **chain** of pairs can be formed in this fashion.

Return `_` the length longest chain which can be formed.

You do not need to use up all the given intervals. You can select pairs in any order.

Example 1:

Input: `pairs = [[1,2],[2,3],[3,4]]` **Output:** `2` **Explanation:** The longest chain is `[1,2] -> [3,4]`.

Example 2:

Input: `pairs = [[1,2],[7,8],[4,5]]` **Output:** `3` **Explanation:** The longest chain is `[1,2] -> [4,5] -> [7,8]`.

Constraints:

`n == pairs.length` `1 <= n <= 1000` `-1000 <= lefti < righti <= 1000`

Code Snippets

C++:

```
class Solution {  
public:  
    int findLongestChain(vector<vector<int>>& pairs) {  
  
    }  
};
```

Java:

```
class Solution {  
    public int findLongestChain(int[][] pairs) {  
  
    }  
}
```

Python3:

```
class Solution:  
    def findLongestChain(self, pairs: List[List[int]]) -> int:
```