

Problem 2140: Solving Questions With Brainpower

Problem Information

Difficulty: Medium

Acceptance Rate: 60.25%

Paid Only: No

Tags: Array, Dynamic Programming

Problem Description

You are given a **0-indexed** 2D integer array `questions` where `questions[i] = [points_i, brainpower_i]`.

The array describes the questions of an exam, where you have to process the questions **in order** (i.e., starting from question `0`) and make a decision whether to **solve** or **skip** each question. Solving question `i` will **earn** you `points_i` points but you will be **unable** to solve each of the next `brainpower_i` questions. If you skip question `i`, you get to make the decision on the next question.

* For example, given `questions = [[3, 2], [4, 3], [4, 4], [2, 5]]`:
* If question `0` is solved, you will earn `3` points but you will be unable to solve questions `1` and `2`. * If instead, question `0` is skipped and question `1` is solved, you will earn `4` points but you will be unable to solve questions `2` and `3`.

Return _the**maximum** points you can earn for the exam_.

Example 1:

Input: questions = [[3,2],[4,3],[4,4],[2,5]] **Output:** 5 **Explanation:** The maximum points can be earned by solving questions 0 and 3. - Solve question 0: Earn 3 points, will be unable to solve the next 2 questions - Unable to solve questions 1 and 2 - Solve question 3: Earn 2 points Total points earned: 3 + 2 = 5. There is no other way to earn 5 or more points.

Example 2:

****Input:**** questions = [[1,1],[2,2],[3,3],[4,4],[5,5]] ****Output:**** 7 ****Explanation:**** The maximum points can be earned by solving questions 1 and 4. - Skip question 0 - Solve question 1: Earn 2 points, will be unable to solve the next 2 questions - Unable to solve questions 2 and 3 - Solve question 4: Earn 5 points Total points earned: $2 + 5 = 7$. There is no other way to earn 7 or more points.

****Constraints:****

```
* `1 <= questions.length <= 105` * `questions[i].length == 2` * `1 <= points[i], brainpower[i] <= 105`
```

Code Snippets

C++:

```
class Solution {  
public:  
    long long mostPoints(vector<vector<int>>& questions) {  
  
    }  
};
```

Java:

```
class Solution {  
public long mostPoints(int[][] questions) {  
  
    }  
}
```

Python3:

```
class Solution:  
    def mostPoints(self, questions: List[List[int]]) -> int:
```