Problem 2724: Sort By

Problem Information

Difficulty: Easy
Acceptance Rate: 0.00%
Paid Only: No

Problem Description

Given an array

arr

and a function

fn

, return a sorted array

sortedArr

. You can assume

fn

only returns numbers and those numbers determine the sort order of

sortedArr

.

sortedArr

must be sorted in

ascending order

by

fn

output.

You may assume that

fn

will never duplicate numbers for a given array.

Example 1:

Input:

arr = [5, 4, 1, 2, 3], fn = (x) => x

Output:

[1, 2, 3, 4, 5]

Explanation:

fn simply returns the number passed to it so the array is sorted in ascending order.

Example 2:

Input:

arr = [{"x": 1}, {"x": 0}, {"x": -1}], fn = (d) => d.x

Output:

[{"x": -1}, {"x": 0}, {"x": 1}]

Explanation:

fn returns the value for the "x" key. So the array is sorted based on that value.

Example 3:

Input:

arr = [[3, 4], [5, 2], [10, 1]], fn = (x) => x[1]

Output:

[[10, 1], [5, 2], [3, 4]]

Explanation:

arr is sorted in ascending order by number at index=1.

Constraints:

arr

is a valid JSON array

fn

is a function that returns a number

1 <= arr.length <= 5 * 10

5

Code Snippets

JavaScript:

```
/**
 * @param {Array} arr
 * @param {Function} fn
 * @return {Array}
```

```
*/
var sortBy = function(arr, fn) {

};
```

TypeScript:

```typescript
type JSONValue = null | boolean | number | string | JSONValue[] | { [key:
string]: JSONValue };
type Fn = (value: JSONValue) => number

function sortBy(arr: JSONValue[], fn: Fn): JSONValue[] {

};
```

Solutions

JavaScript Solution:

```javascript
/**
 * Problem: Sort By
 * Difficulty: Easy
 * Tags: array, sort
 *
 * Approach: Use two pointers or sliding window technique
 * Time Complexity: O(n) or O(n log n)
 * Space Complexity: O(1) to O(n) depending on approach
 */

/**
 * @param {Array} arr
 * @param {Function} fn
 * @return {Array}
 */
var sortBy = function(arr, fn) {

};
```

TypeScript Solution:

```
/**
 * Problem: Sort By
 * Difficulty: Easy
 * Tags: array, sort
 *
 * Approach: Use two pointers or sliding window technique
 * Time Complexity: O(n) or O(n log n)
 * Space Complexity: O(1) to O(n) depending on approach
 */


type JSONValue = null | boolean | number | string | JSONValue[] | { [key:
string]: JSONValue };
type Fn = (value: JSONValue) => number


function sortBy(arr: JSONValue[], fn: Fn): JSONValue[] {

};
```