

Problem 2618: Check if Object Instance of Class

Problem Information

Difficulty: Medium

Acceptance Rate: 0.00%

Paid Only: No

Problem Description

Write a function that checks if a given value is an instance of a given class or superclass. For this problem, an object is considered an instance of a given class if that object has access to that class's methods.

There are no constraints on the data types that can be passed to the function. For example, the value or the class could be

undefined

.

Example 1:

Input:

```
func = () => checkIfInstanceOf(new Date(), Date)
```

Output:

true

Explanation:

The object returned by the Date constructor is, by definition, an instance of Date.

Example 2:

Input:

```
func = () => { class Animal {}; class Dog extends Animal {}; return checkIfInstanceOf(new Dog(), Animal); }
```

Output:

true

Explanation:

```
class Animal {}; class Dog extends Animal {}; checkIfInstanceOf(new Dog(), Animal); // true
```

Dog is a subclass of Animal. Therefore, a Dog object is an instance of both Dog and Animal.

Example 3:

Input:

```
func = () => checkIfInstanceOf(Date, Date)
```

Output:

false

Explanation:

A date constructor cannot logically be an instance of itself.

Example 4:

Input:

```
func = () => checkIfInstanceOf(5, Number)
```

Output:

true

Explanation:

5 is a Number. Note that the "instanceof" keyword would return false. However, it is still considered an instance of Number because it accesses the Number methods. For example "toFixed()".

Code Snippets

JavaScript:

```
/**
 * @param {*} obj
 * @param {*} classFunction
 * @return {boolean}
 */
var checkIfInstanceOf = function(obj, classFunction) {

};

/**
 * checkIfInstanceOf(new Date(), Date); // true
 */
```

TypeScript:

```
function checkIfInstanceOf(obj: any, classFunction: any): boolean {

};

/**
 * checkIfInstanceOf(new Date(), Date); // true
 */
```

Solutions

JavaScript Solution:

```
/**
 * Problem: Check if Object Instance of Class
```

```

* Difficulty: Medium
* Tags: general
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

/**
 * @param {*} obj
 * @param {*} classFunction
 * @return {boolean}
 */
var checkIfInstanceOf = function(obj, classFunction) {

};

/**
 * checkIfInstanceOf(new Date(), Date); // true
 */

```

TypeScript Solution:

```

/**
 * Problem: Check if Object Instance of Class
 * Difficulty: Medium
 * Tags: general
 *
 * Approach: Optimized algorithm based on problem constraints
 * Time Complexity: O(n) to O(n^2) depending on approach
 * Space Complexity: O(1) to O(n) depending on approach
 */

function checkIfInstanceOf(obj: any, classFunction: any): boolean {

};

/**
 * checkIfInstanceOf(new Date(), Date); // true
 */

```