Problem

Given a function fn, return a ****memoized**** version of that function.

A ****memoized****function is a function that will never be called twice with the same inputs. Instead it will return a cached value.

You can assume there are ****3****possible input functions: sum****,****fib****,****and factorial****.****

* sumaccepts two integers a and b and returns a + b.
* fibaccepts a single integer n and returns 1 if n <= 1 or fib(n - 1) + fib(n - 2) otherwise.
* factorial accepts a single integer n and returns 1 if n <= 1 or factorial(n - 1) \* n otherwise.

Solution

/\*\*

 \* @anmol {Function} fn

 \* @return {Function}

 \*/

function memoize(fn) {

    const cache = new Map();

    return function(...args) {

        const key = args.toString();

        if (cache.has(key)) {

            return cache.get(key);

        } else {

            const result = fn (...args);

            cache.set(key, result);

            return result

        }

    }

}