



ES6 Collections

Mục tiêu:

- Hiểu được lịch sử của Javascript và phiên bản ES6 của JS
- Hiểu được các feature mới của ES6: Arrow function, Classes, Block scope, Rest/Spread, Destructuring, Template string, Map/Set
- Sử dụng được các cú pháp mới của ES6 để code có thể dễ đọc hơn.

Exercise 01

Sử dụng ES6 Generators để mô tả chức năng của Object Spread trong Javascript

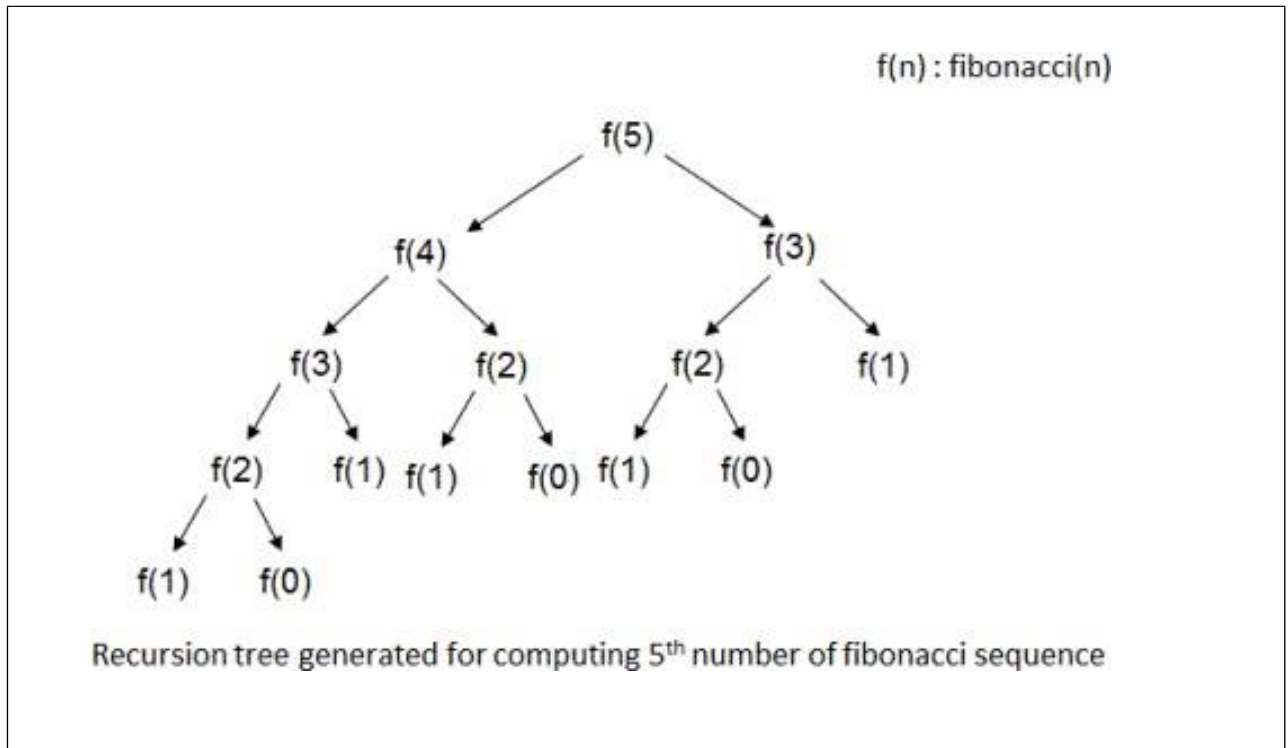
```
1. var numbers = {  
2.   // ..  
3. };  
4.  
5. // should print 0..100 by step 1  
6. // 0 1 2 ... 100  
7. for (let num of numbers) {  
8.   console.log(num);  
9. }  
10.  
11. // should print 6..30 by step 4  
12. // 6 10 14 ... 30  
13. for (let num of /*..*/) {  
14.   console.log(num);  
15. }
```

Exercise 02

Cho đoạn code sau có chức năng tính toán dãy fibonacci của một số n:

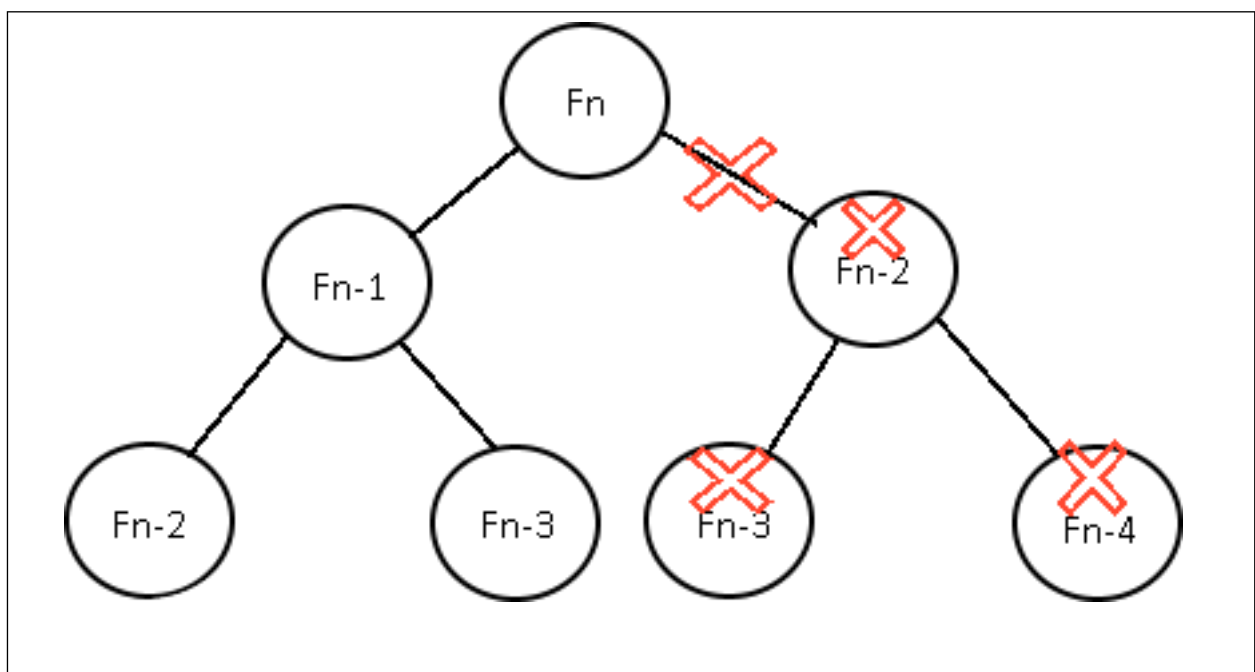
```
1. function fibonacci(n) {  
2.   if (n <= 1) {  
3.     return 1;  
4.   }  
5.  
6.   return fibonacci(n - 1) + fibonacci(n - 2);  
7. }
```

Vấn đề của đoạn code này là chúng ta lặp đi lặp lại quá nhiều lần tính toán:



Trong hình trên, ta có thể thấy rằng với tham số 2 truyền vào $f(2)$ được thực thi tới 3 lần, $f(3)$ được thực thi 2 lần

Nhiệm vụ của bạn là sử dụng kiểu dữ liệu Map để nhớ được kết quả của Fibonacci của giá trị nhỏ hơn để chúng ta không phải tính toán lại chúng



Ví dụ: chúng ta đã tính toán giá trị fibonacci của $n - 2$ và $n - 3$ ở nhánh bên trái nên không cần phải tính toán lại chúng ở nhánh bên phải nữa

Problem 03

Cho một array --- Một danh sách các đối tượng ở bên
trong

Mỗi đối tượng đều có các thuộc tính: **id, name, age**.

Nhiệm vụ của bạn là xoá đi những đối tượng trùng lặp trong danh sách đó. Mỗi đối tượng bên trong được coi là trùng lặp khi chúng có trùng id

Example:

```
1. [  
2.   {  
3.     id: 1,  
4.     name: 'Dung',  
5.     age: 20  
6.   },  
7.   {  
8.     id: 2,  
9.     name: 'Diu',  
10.    age: 20  
11.  },  
12.  {  
13.    id: 3,  
14.    name: 'Ky',  
15.    age: 20  
16.  },  
17.  {  
18.    id: 1,  
19.    name: 'Hai',  
20.    age: 22  
21.  }  
22. ]
```

Ví dụ, id = 1 bị trùng, nên chúng ta sẽ giữ lại đối tượng với id = 1 và name = 'Dung'

Expected Output:

```
1. [  
2.   {  
3.     id: 1,  
4.     name: 'Dung',  
5.     age: 20  
6.   },  
7.   {  
8.     id: 2,  
9.     name: 'Diu',  
10.    age: 20  
11.  },  
12.  {  
13.    id: 3,  
14.    name: 'Ky',  
15.    age: 20  
16.  }  
17. ]
```
