#CátedrasCiber

Módulo IV: Fuerza bruta y programación dinámica

12/03/2025

Financiado por la Unión Europea
NextGenerationEU

GOBIERNO DE ESPAÑA
MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA
SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL

Plan de Recuperación, Transformación y Resiliencia

Entidad colaboradora:
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD
UC Universidad de Cantabria

Índice del módulo

1. Algoritmos de fuerza bruta.
2. Programación dinámica.
3. Ejercicios de programación dinámica.
4. Cómo crear un formulario.
5. Consejos para los concursos.

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Algoritmos de fuerza bruta

Algoritmos de fuerza bruta

- Recursividad – DFS - BFS

Generar todas las posibles opciones

Ejemplo: programa recursivo de Fibonacci

Sucesión de Fibonacci: 1, 1, 2, 3, 5, 8, …

- Fibonacci(n) =
- 1, si n==0 o n==1
- Fibonacci(n-1) + Fibonacci(n-2), en otro caso

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Algoritmos de fuerza bruta

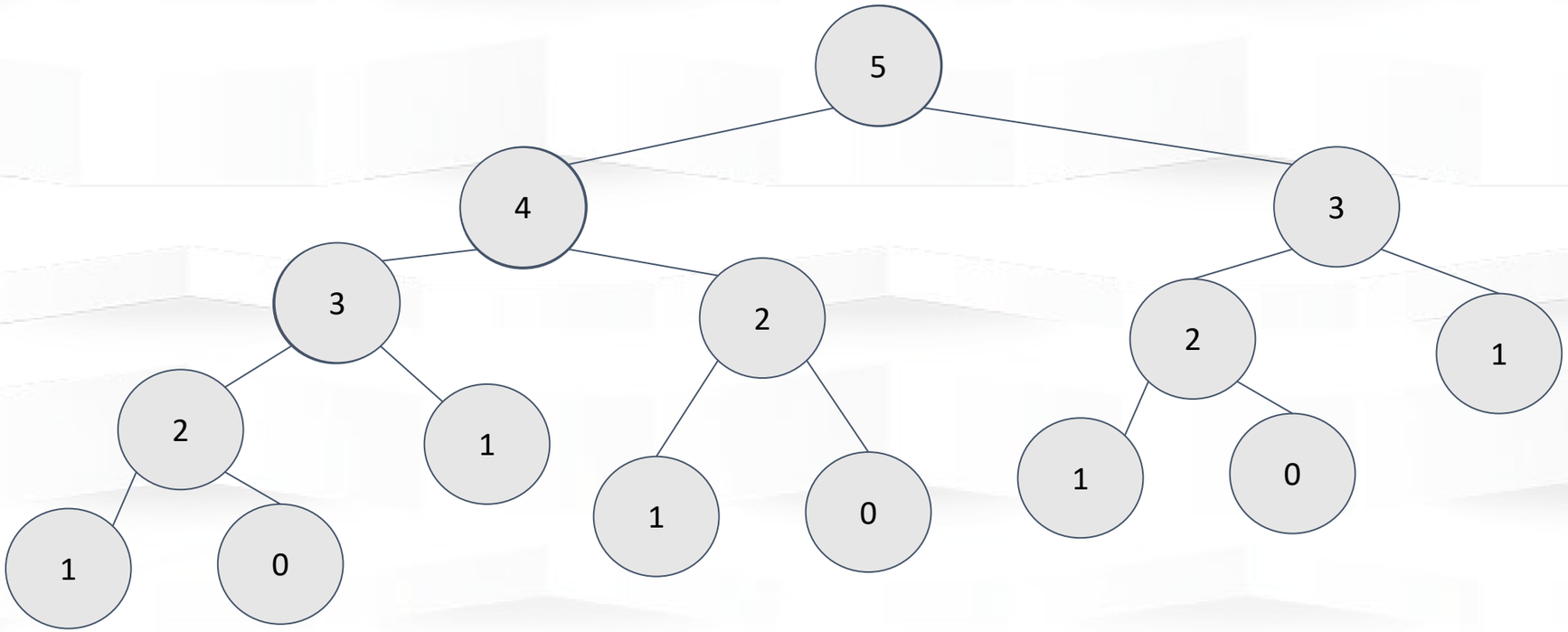
- Ejemplos en ciberseguridad:
  - Crackeo de Hashes
  - Fuerza Bruta en Logins
  - Desencriptación de Datos
  - Inyecciones SQL (Time-Based SQL Injection)

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

Programación dinámica

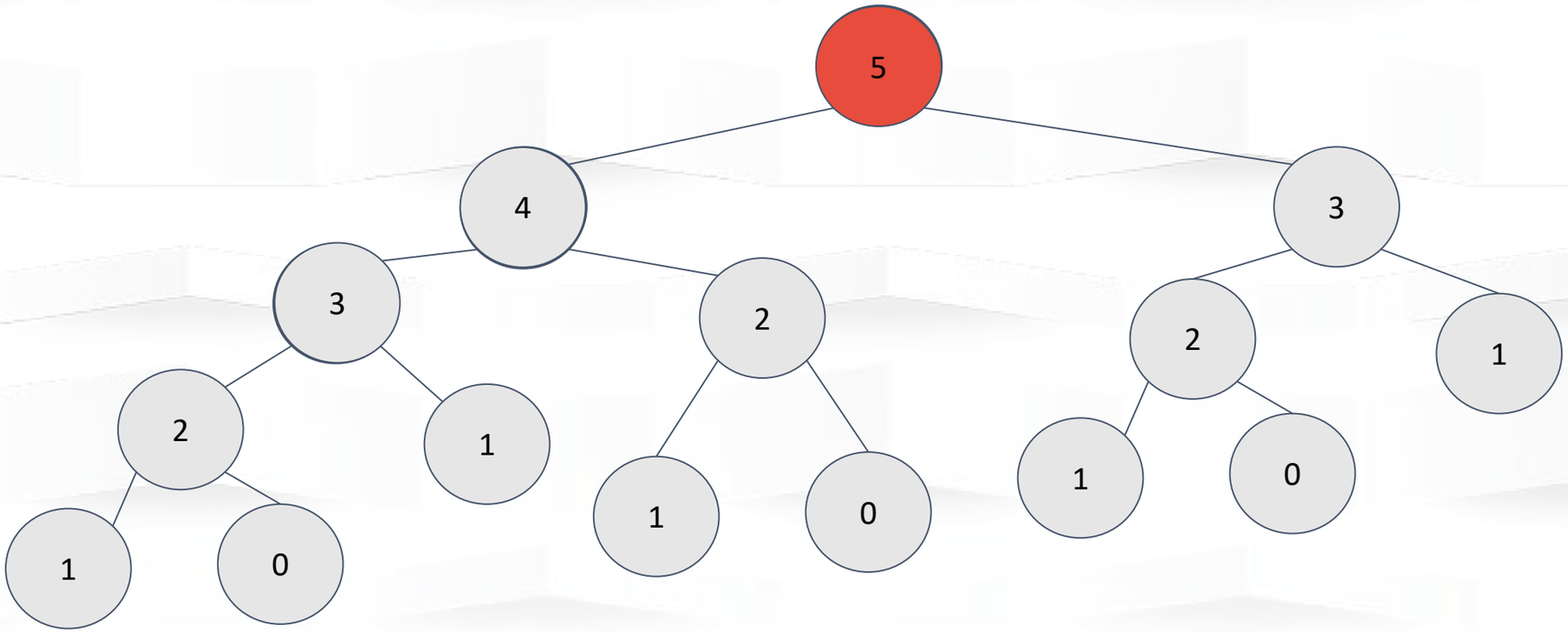
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

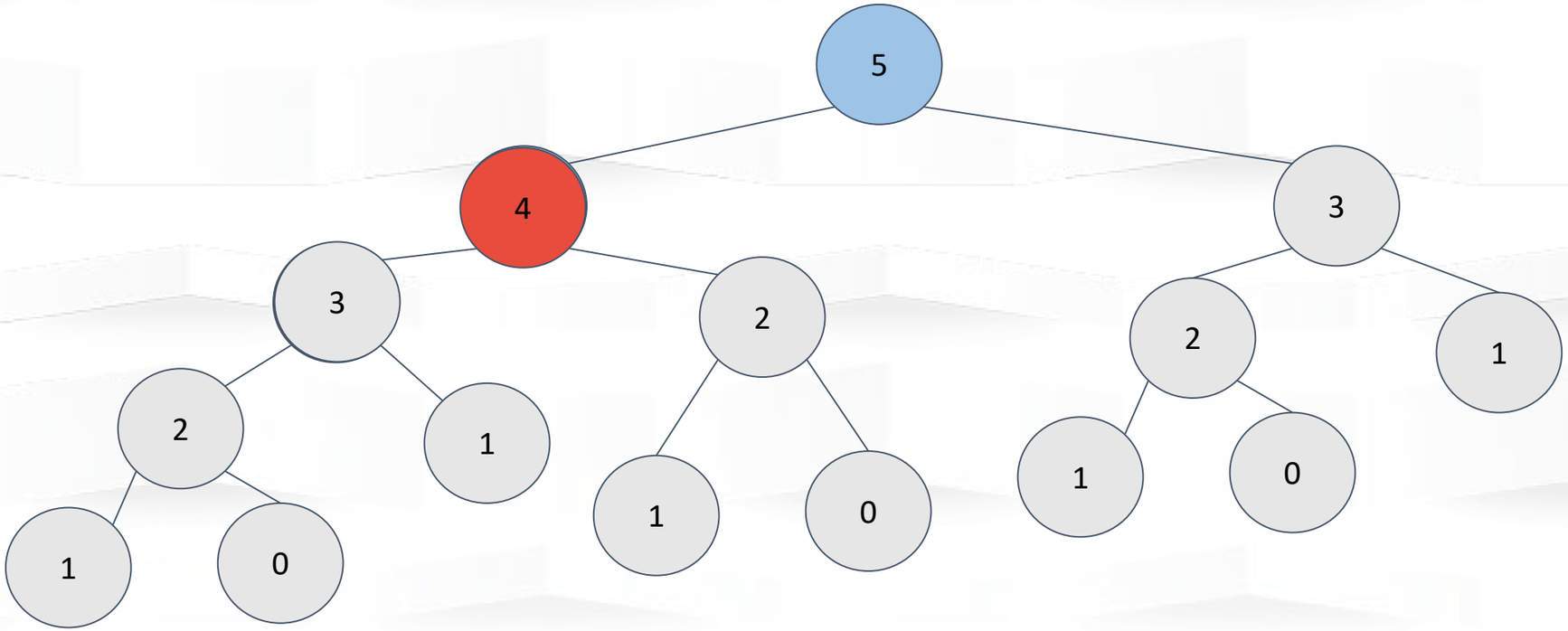
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

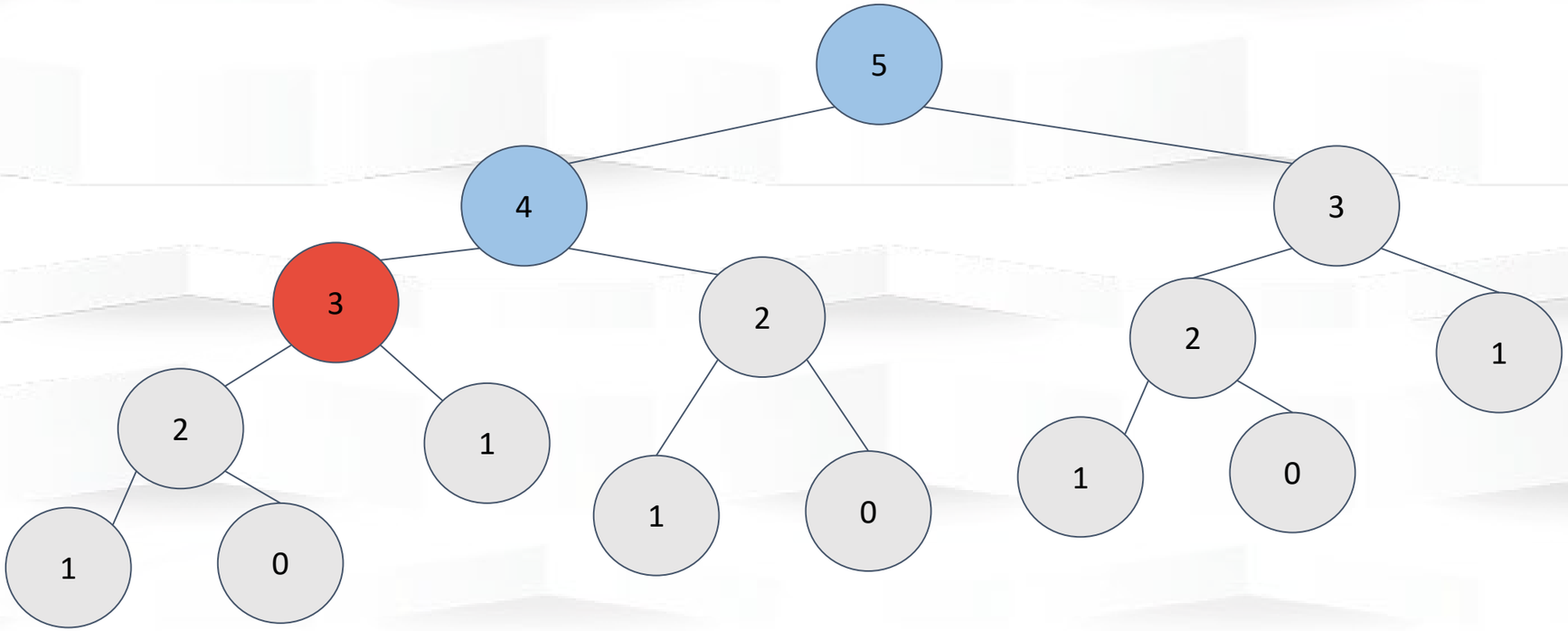
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

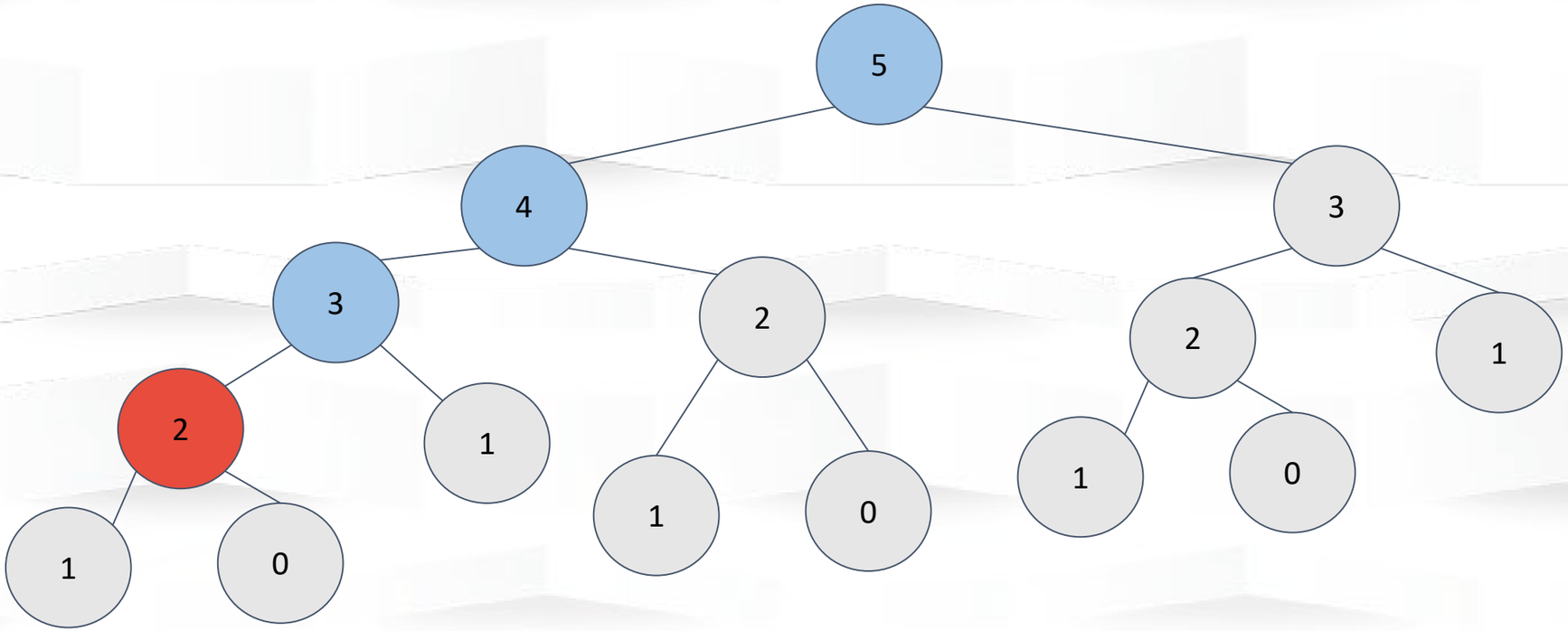
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

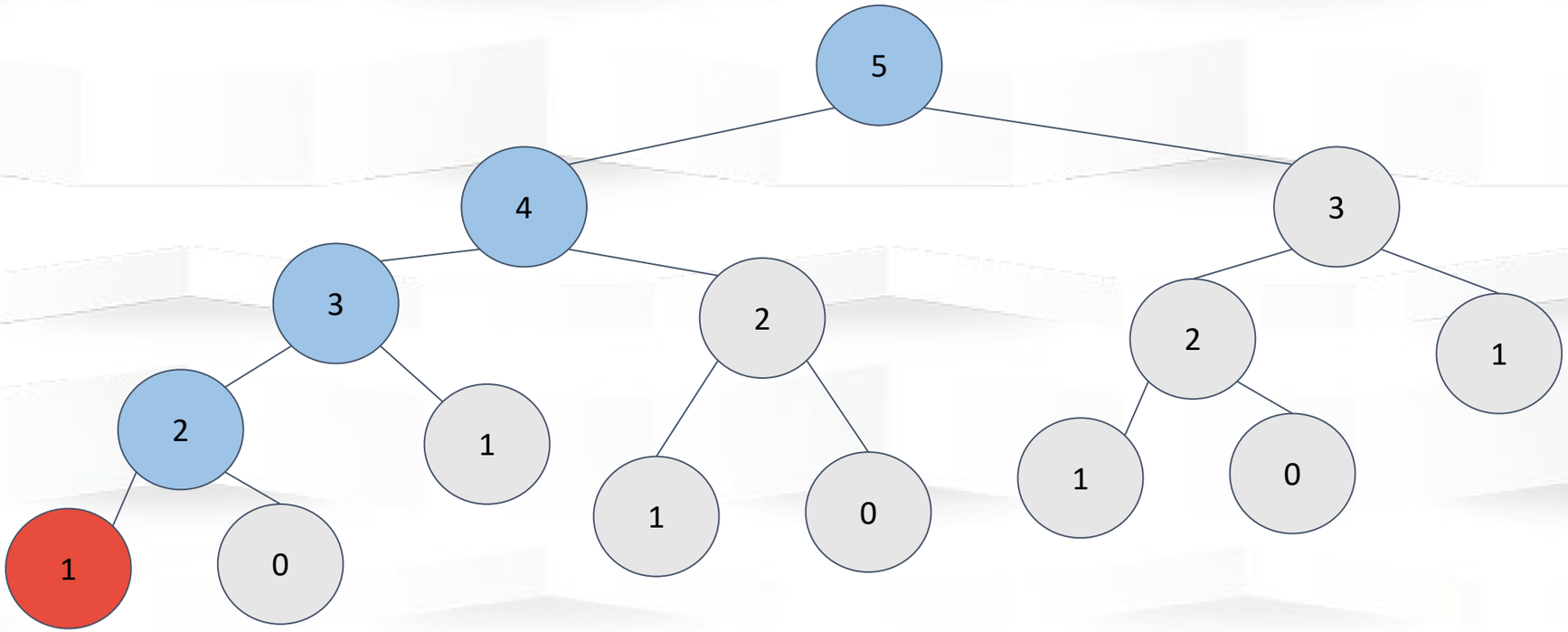
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

Ejemplo: Fibonacci(5)

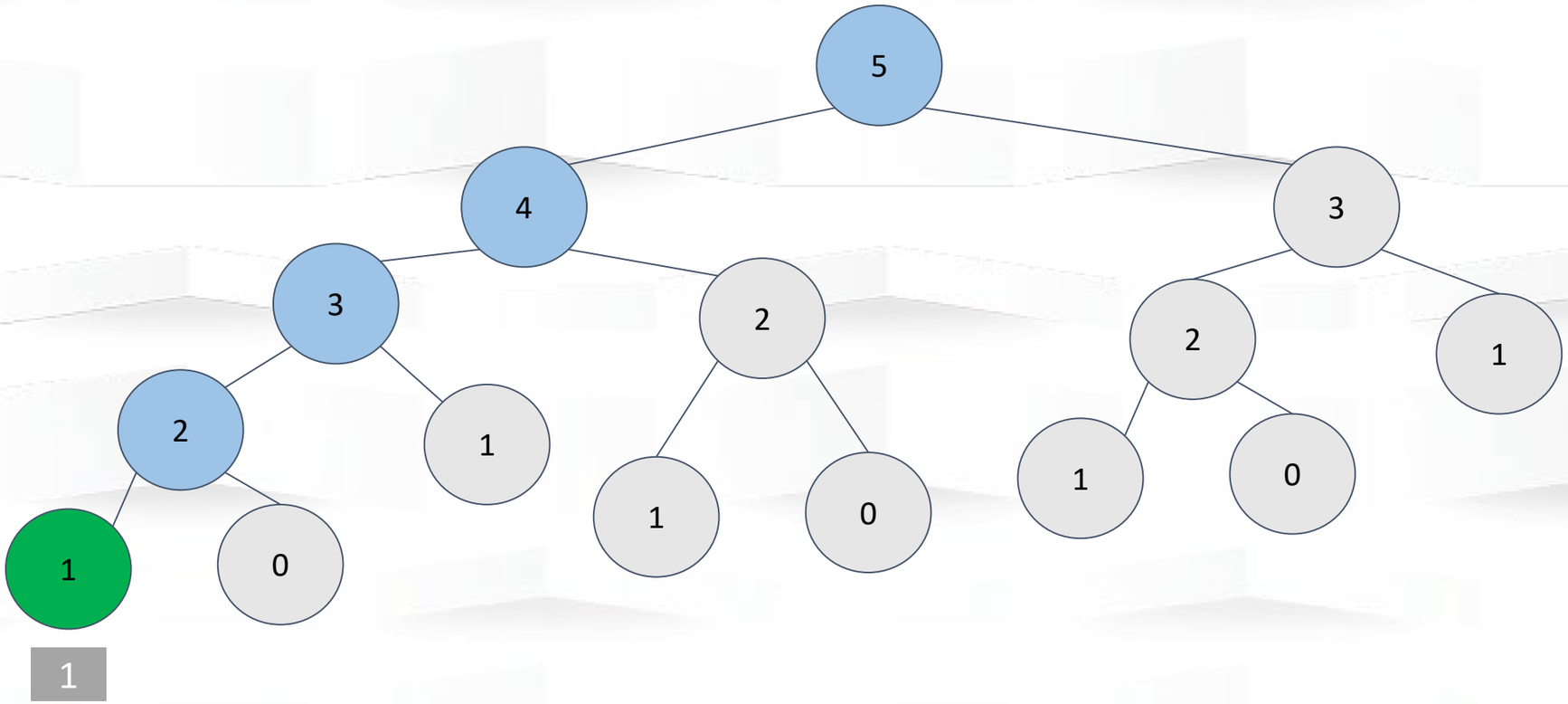


Caso base: Fibonacci(1) = 1

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

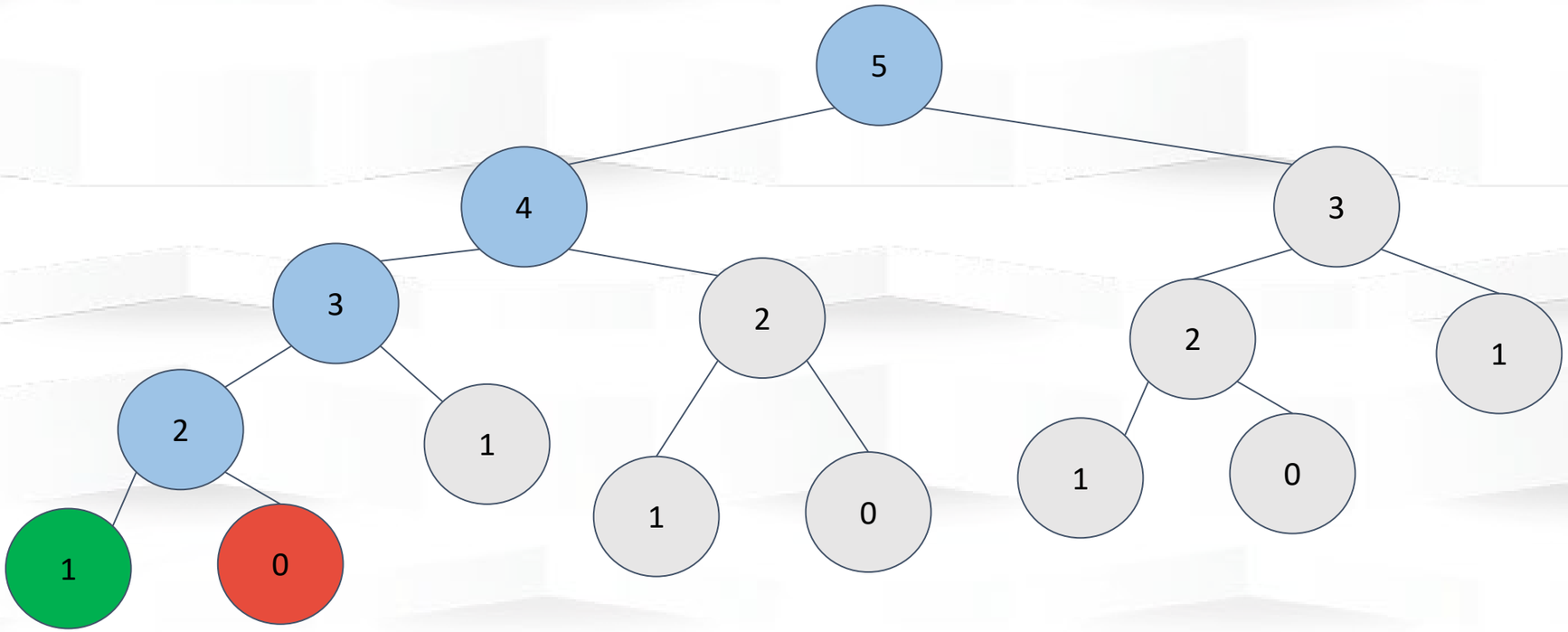
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

Ejemplo: Fibonacci(5)



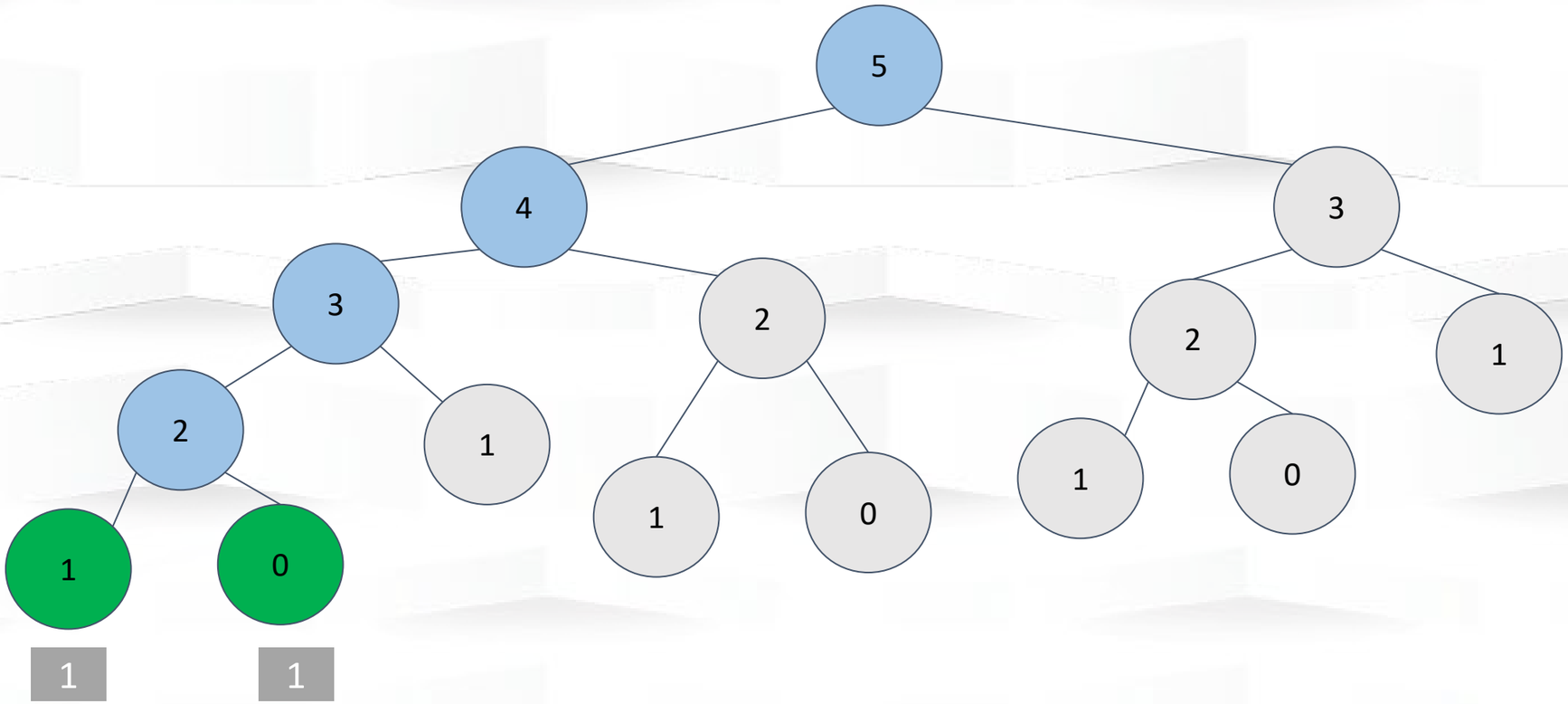
1

Caso base: Fibonacci(0) = 1

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

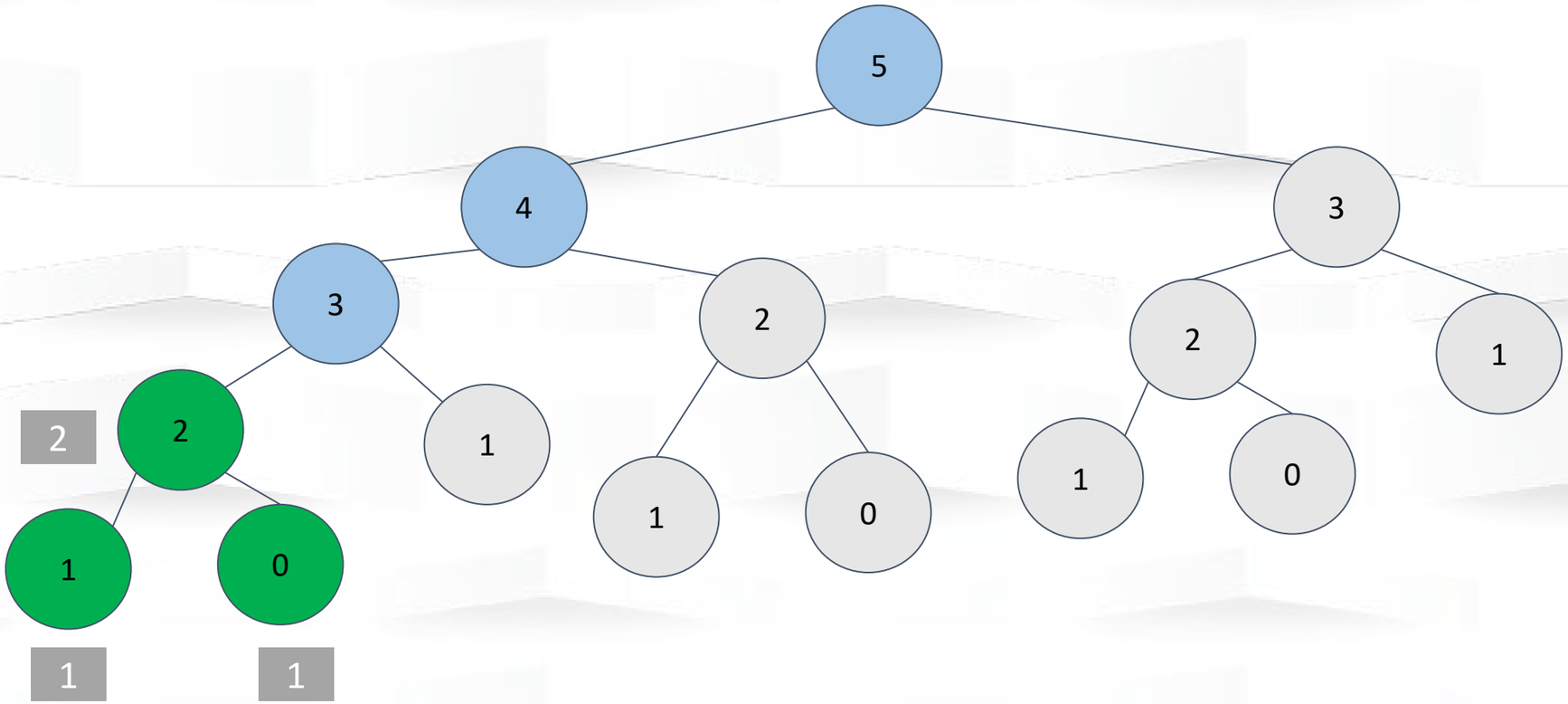
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

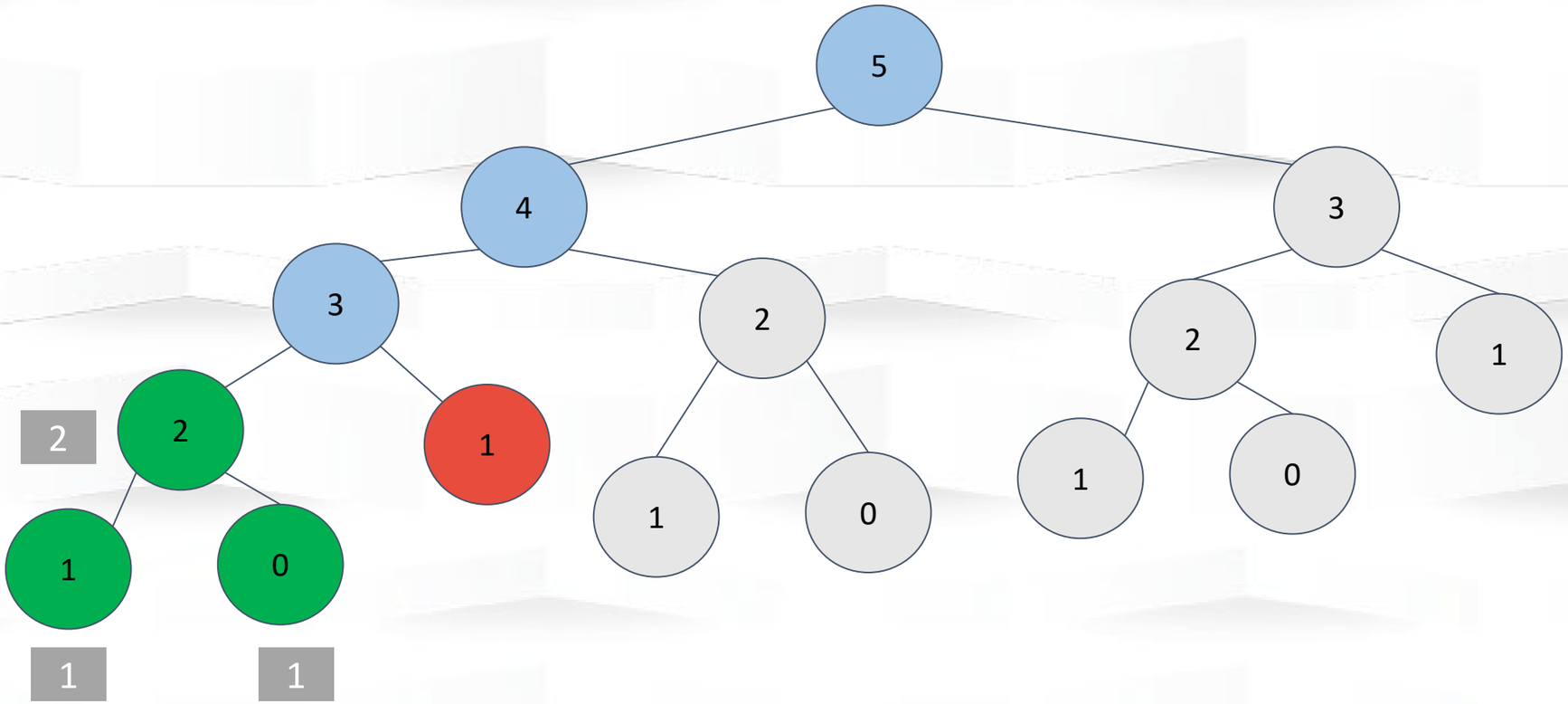
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

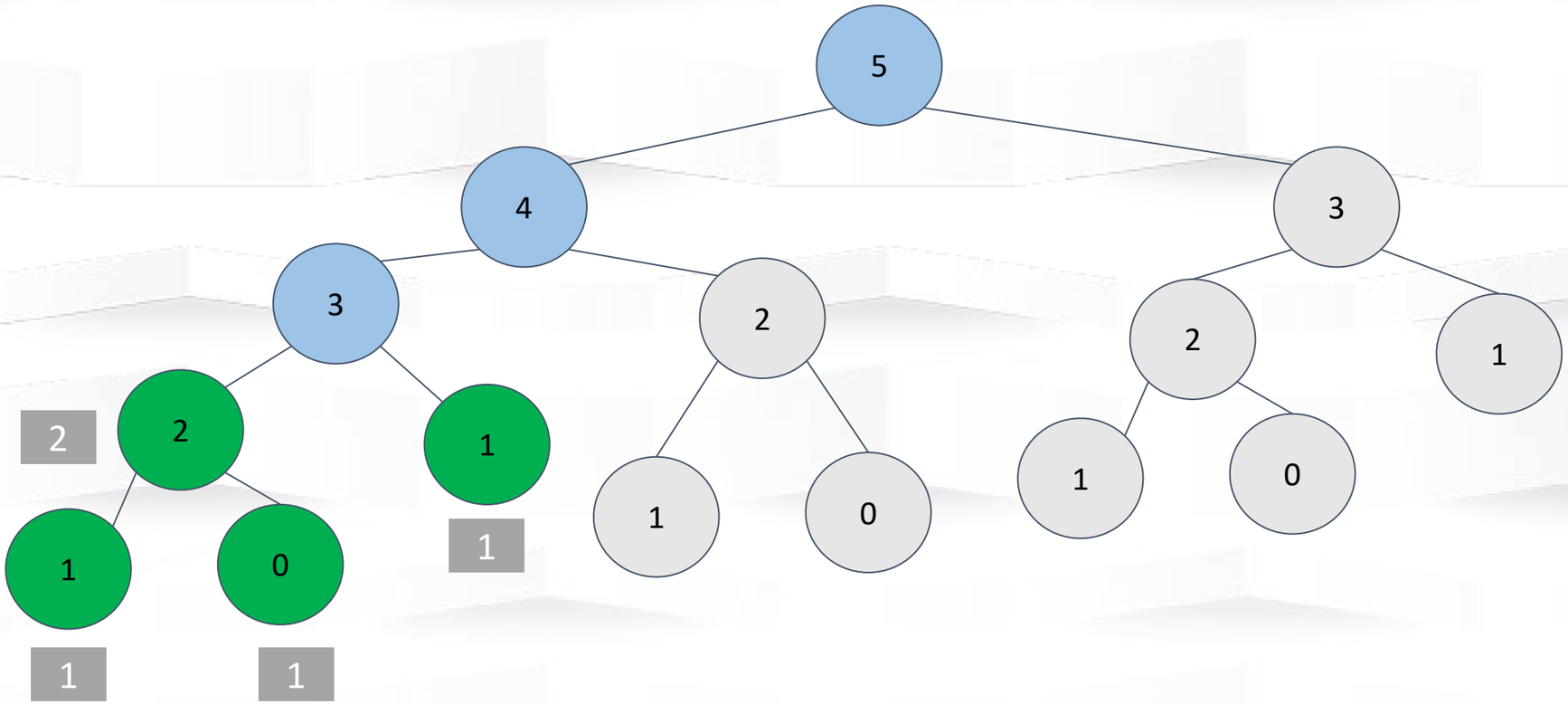
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

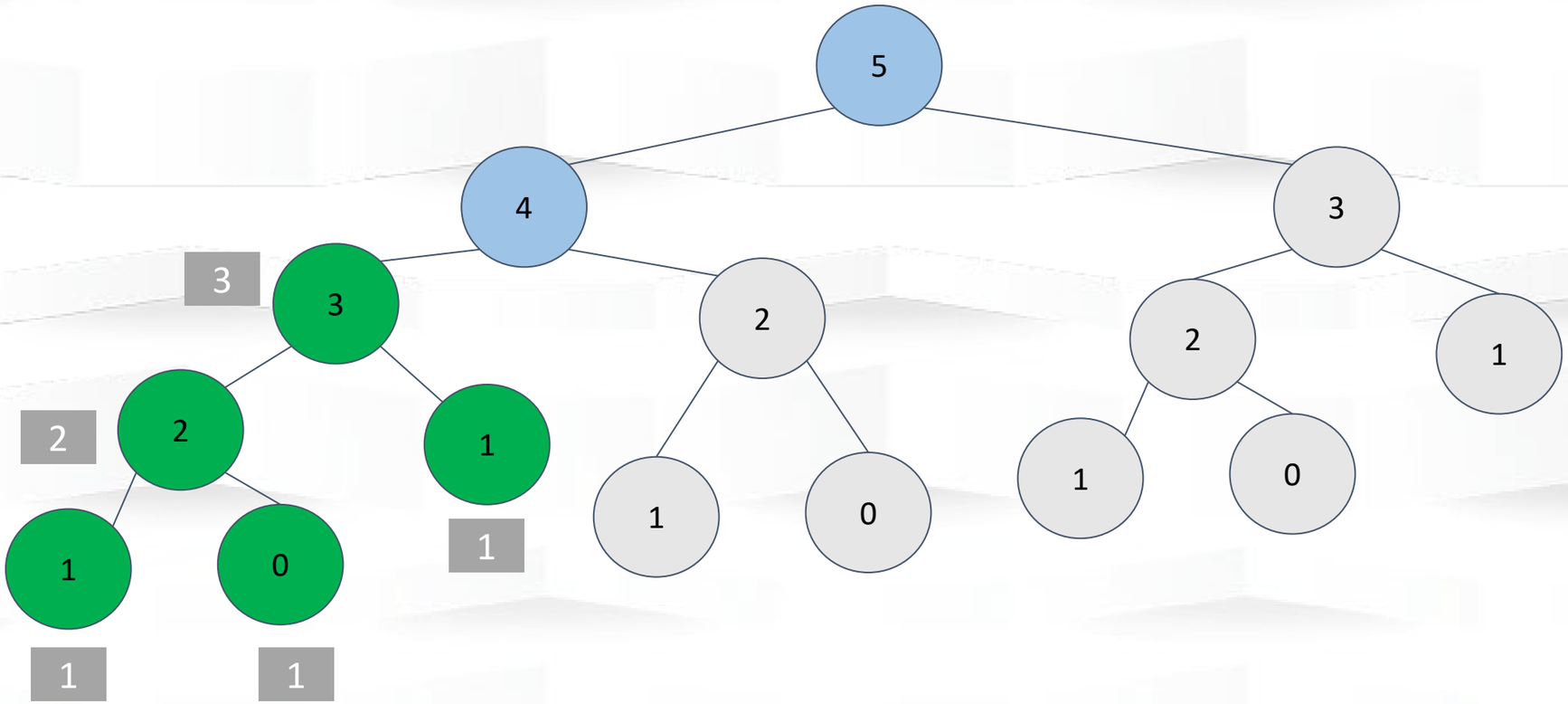
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

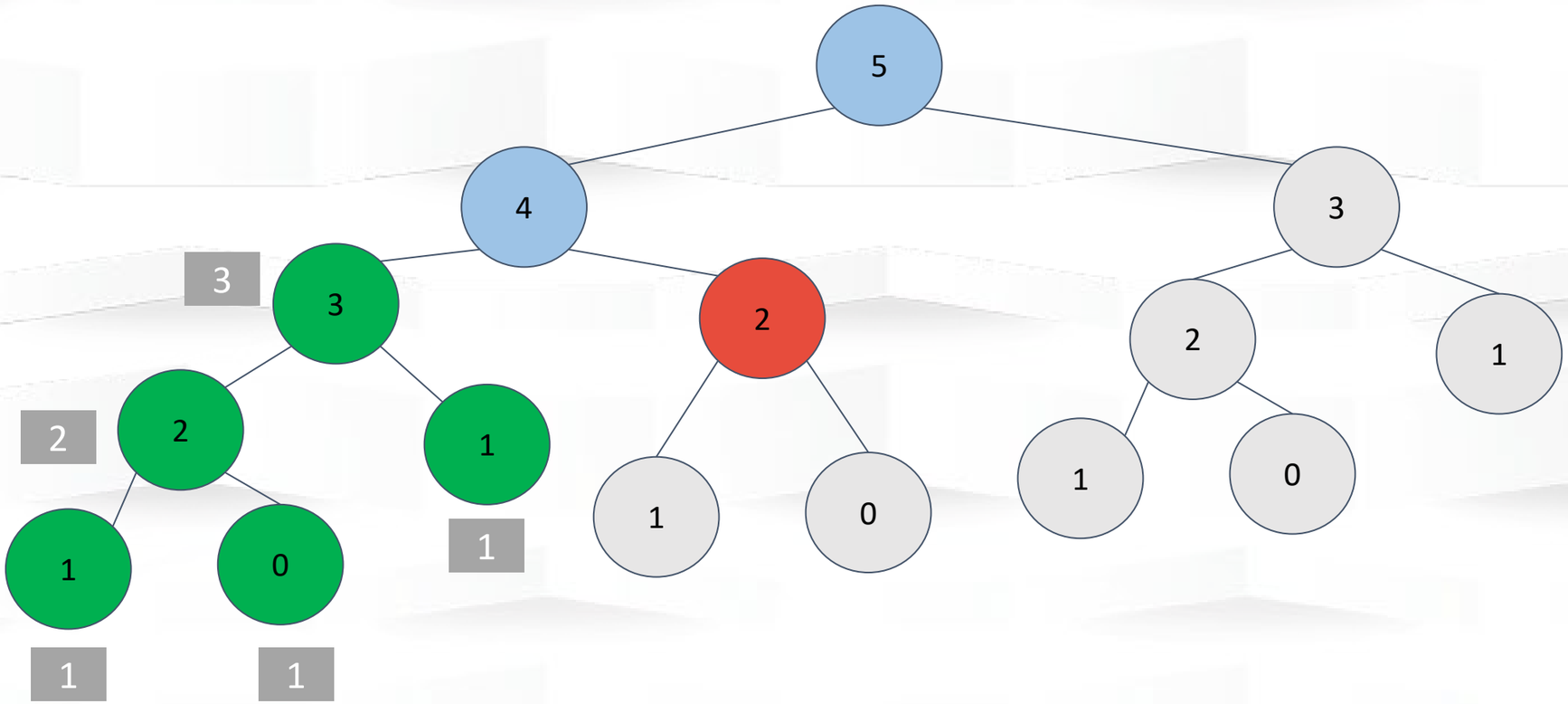
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

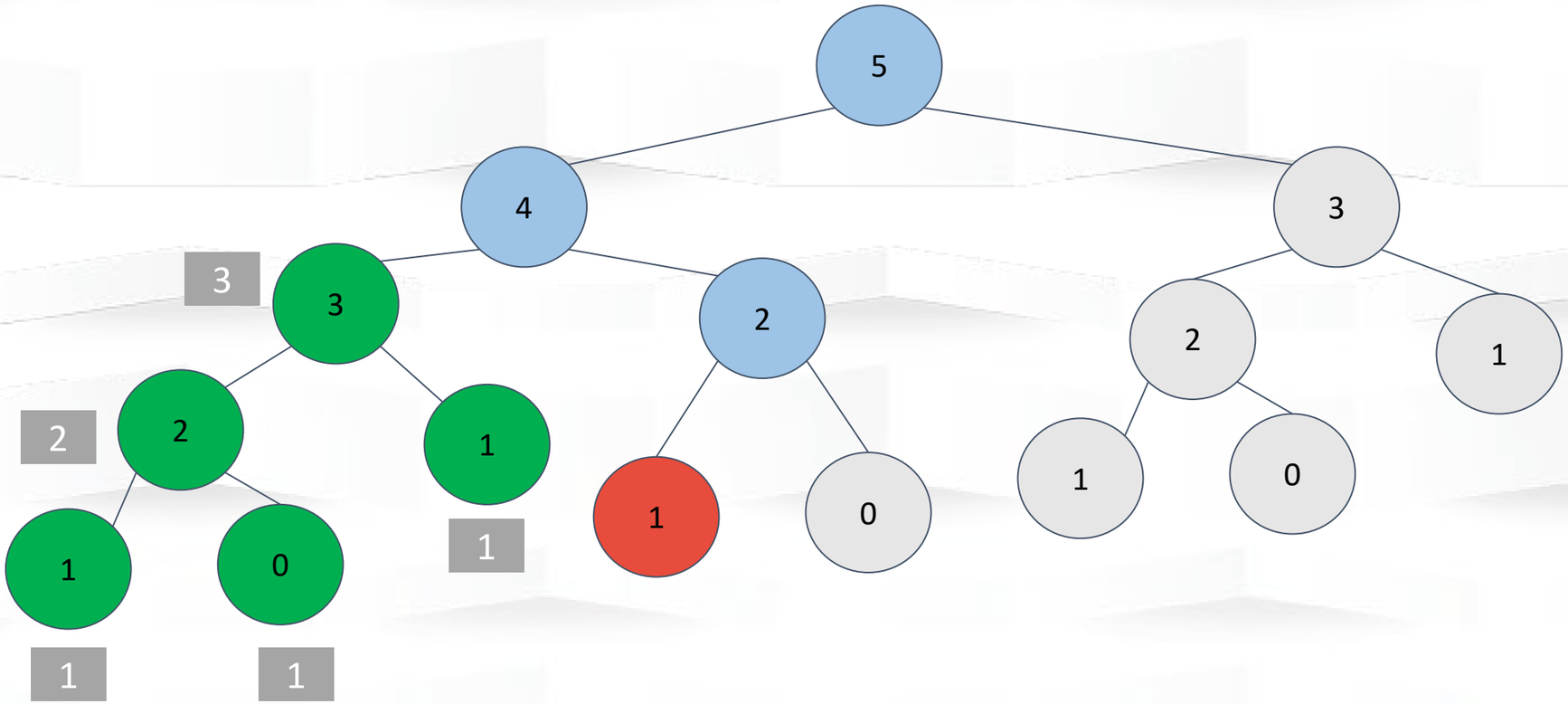
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

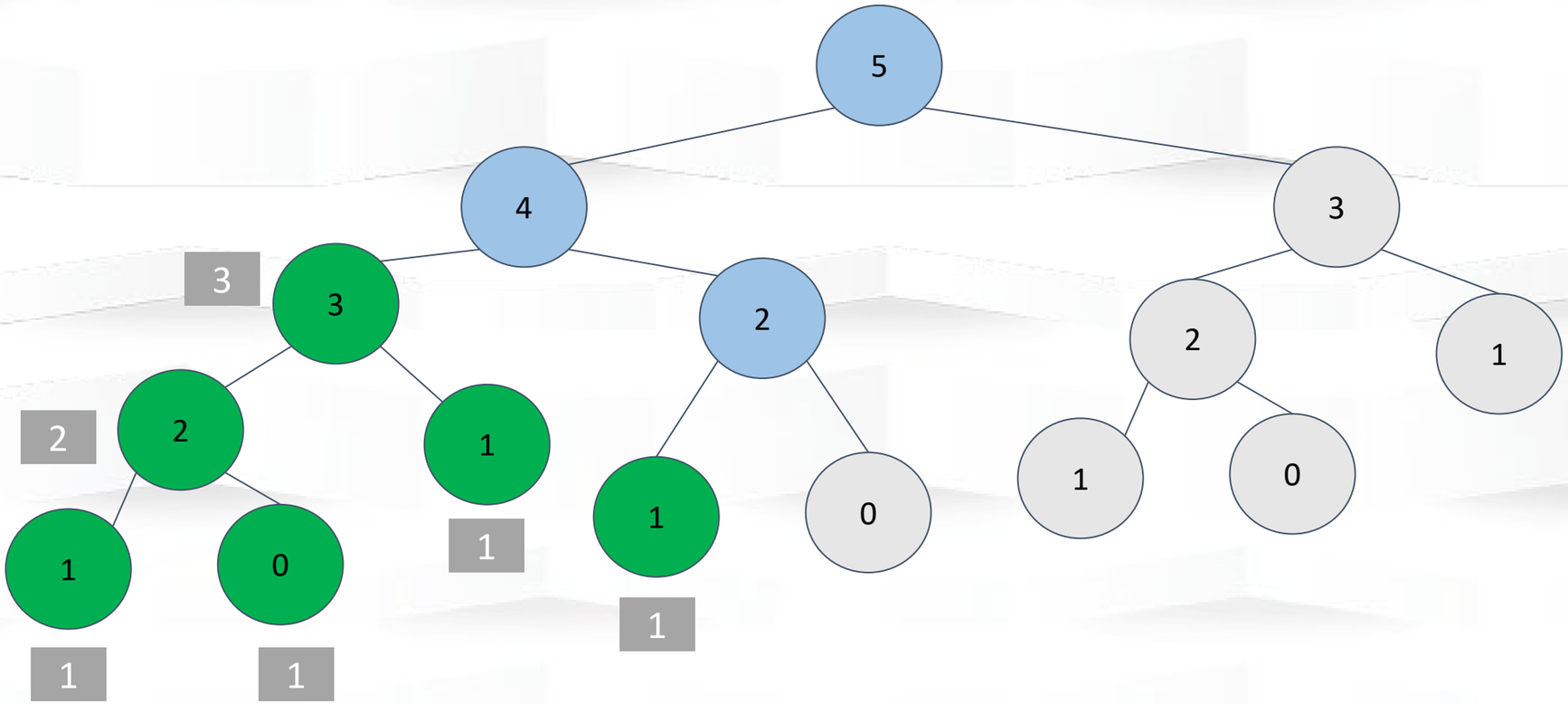
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

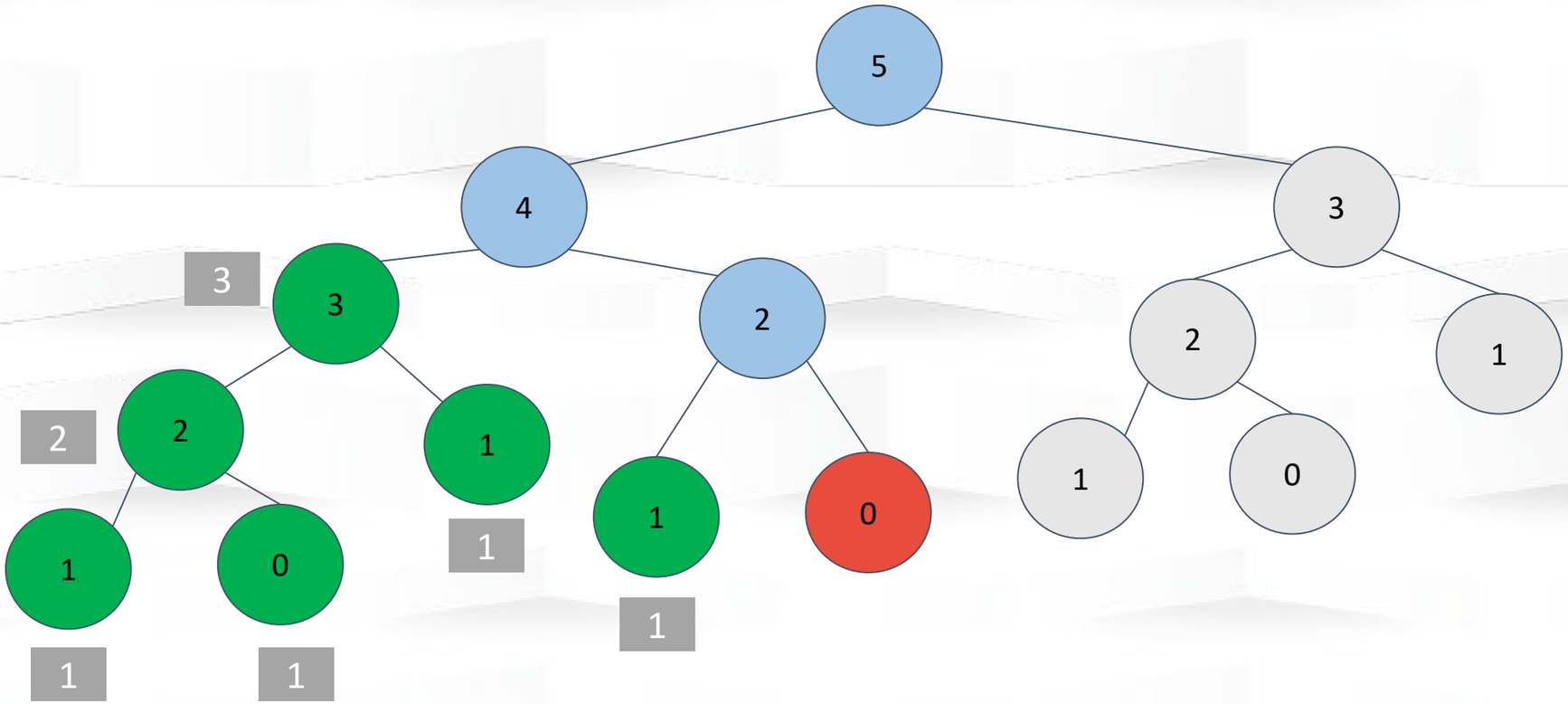
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

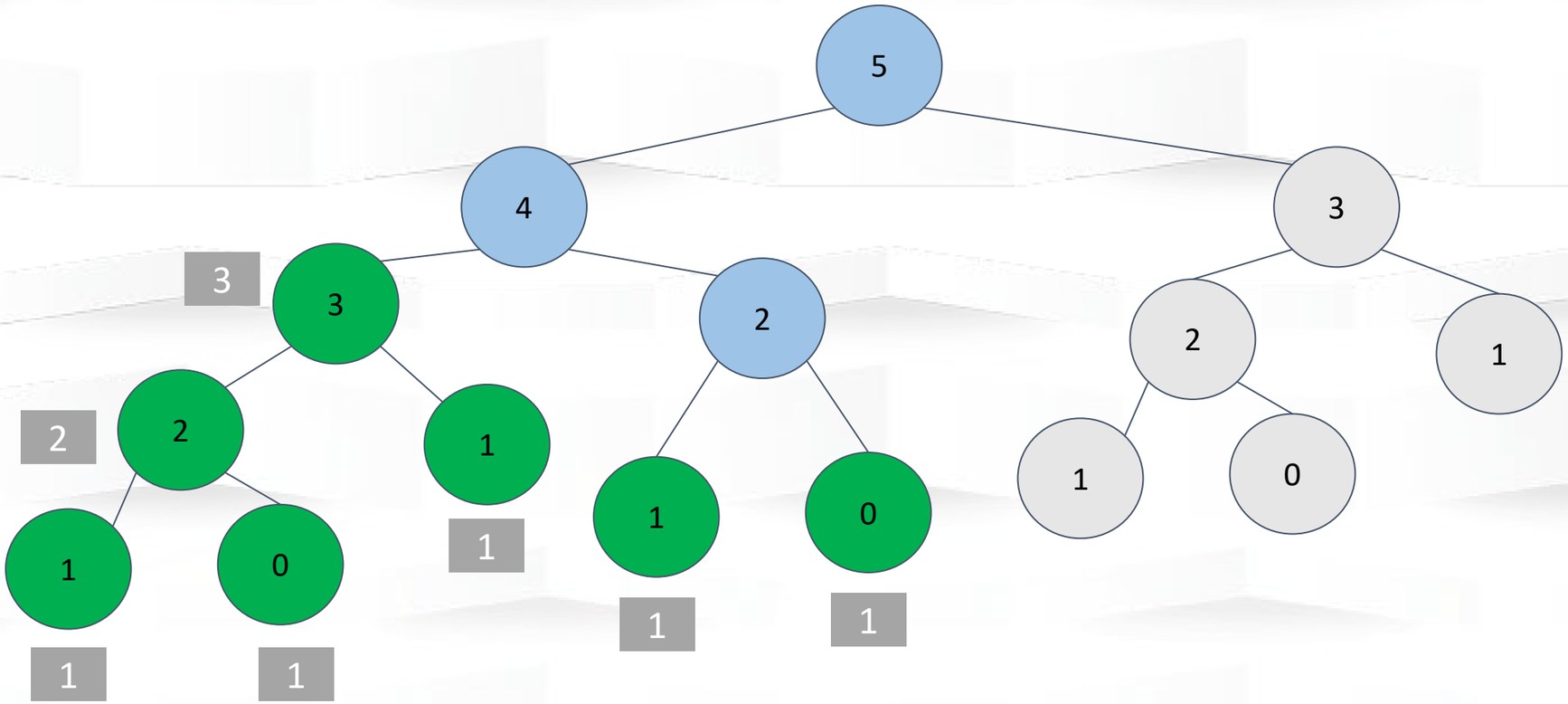
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

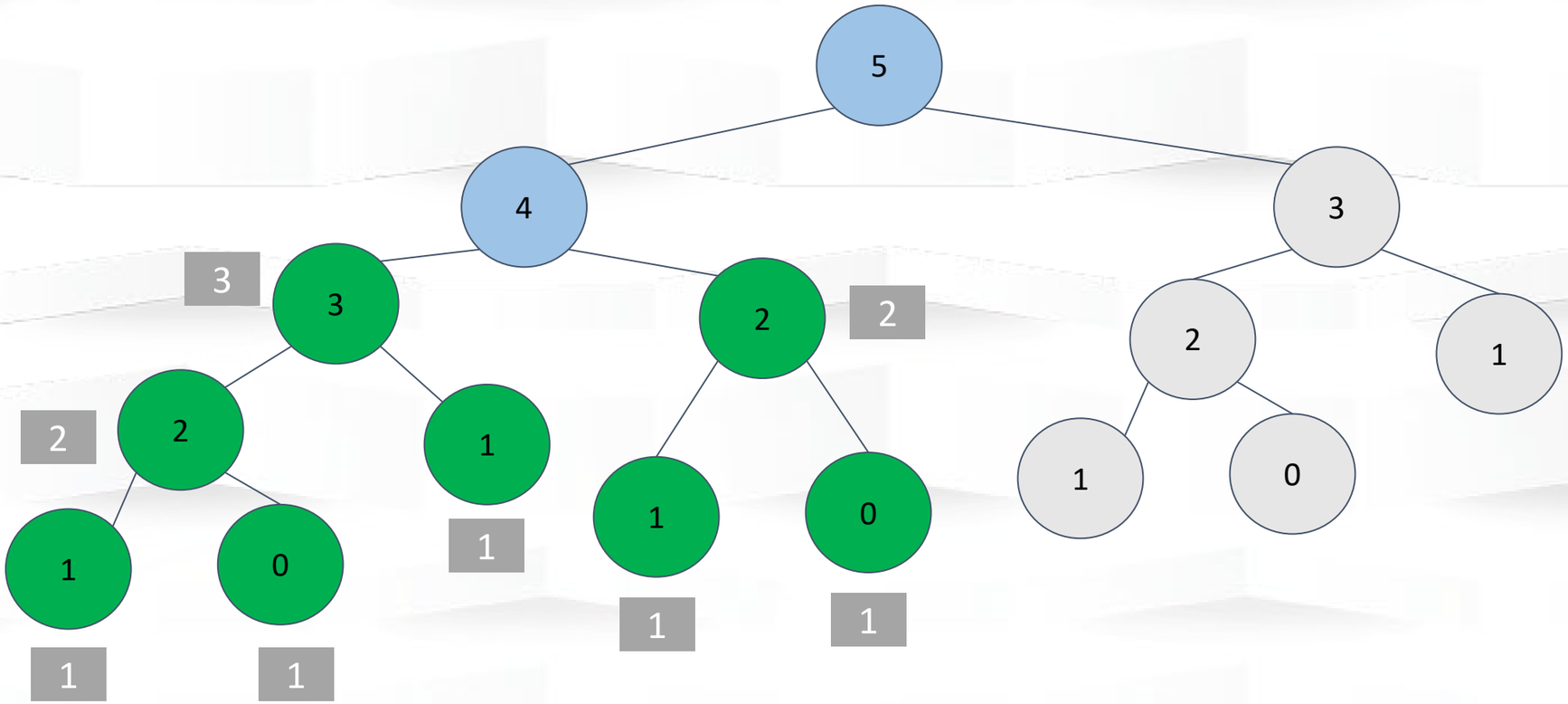
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

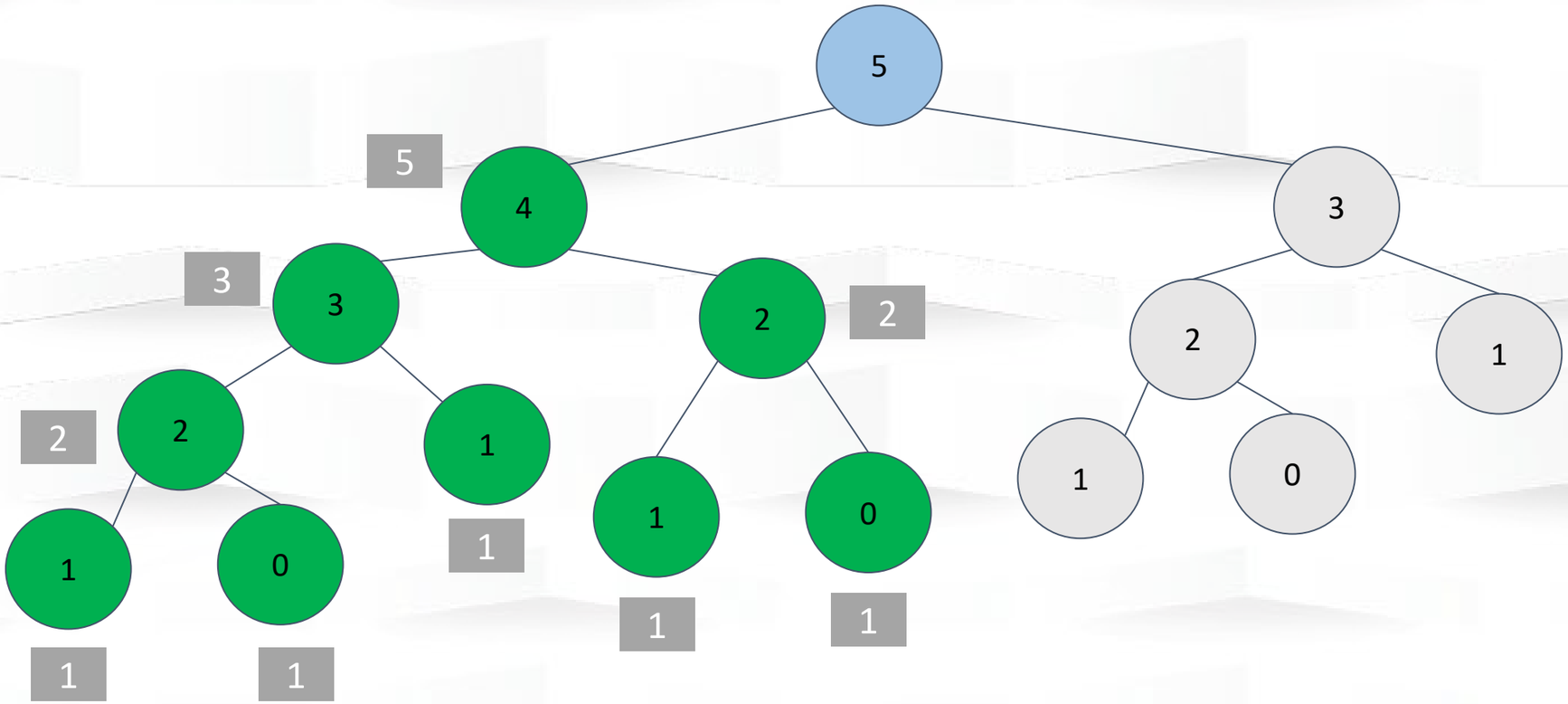
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

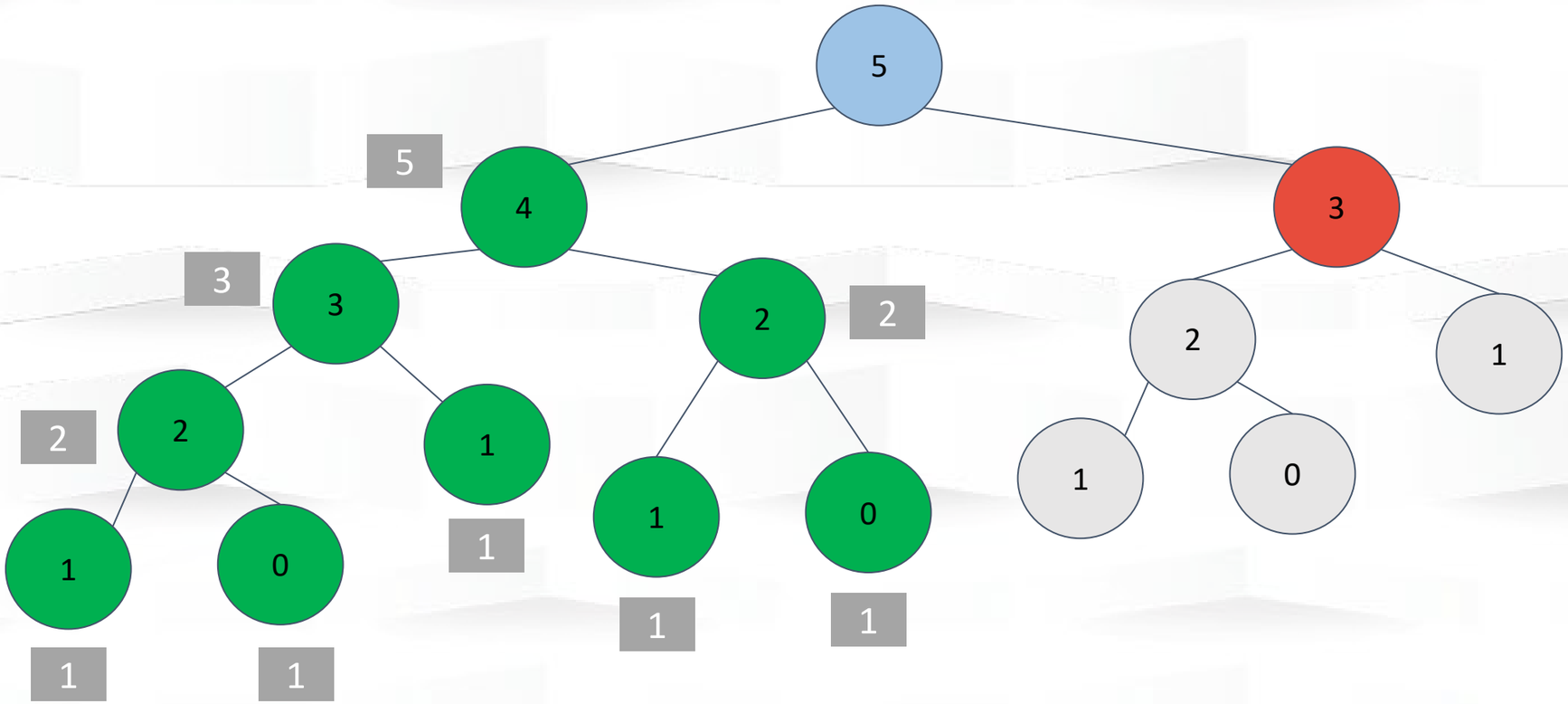
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

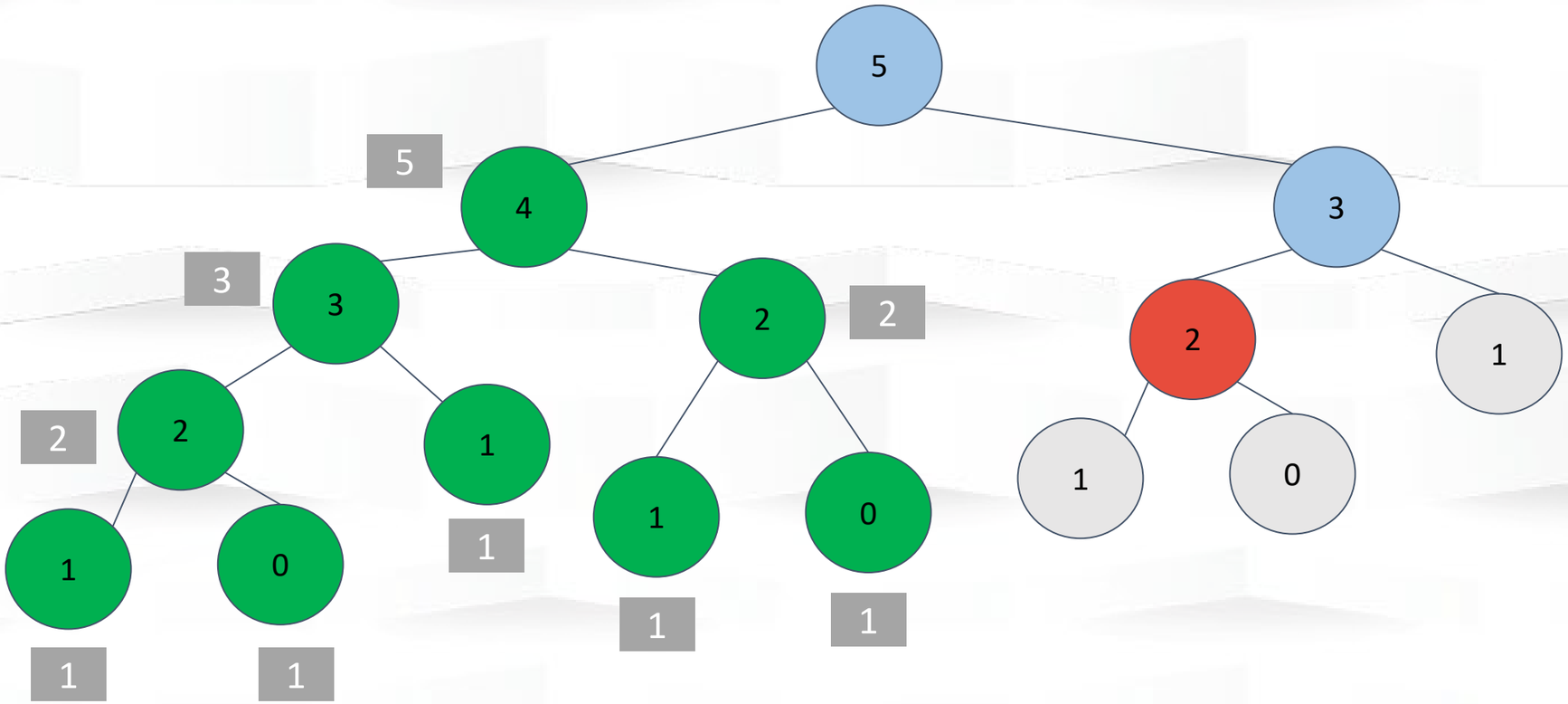
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

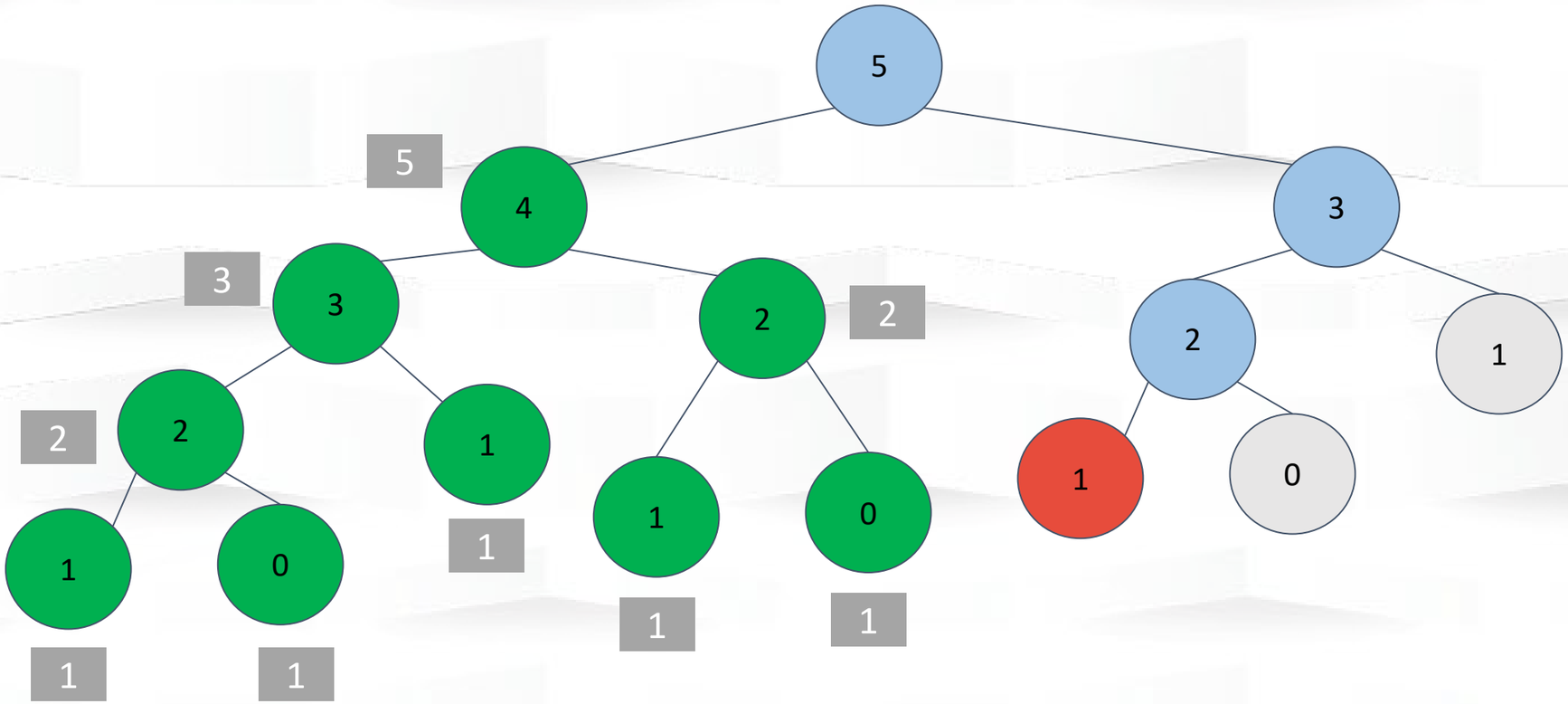
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

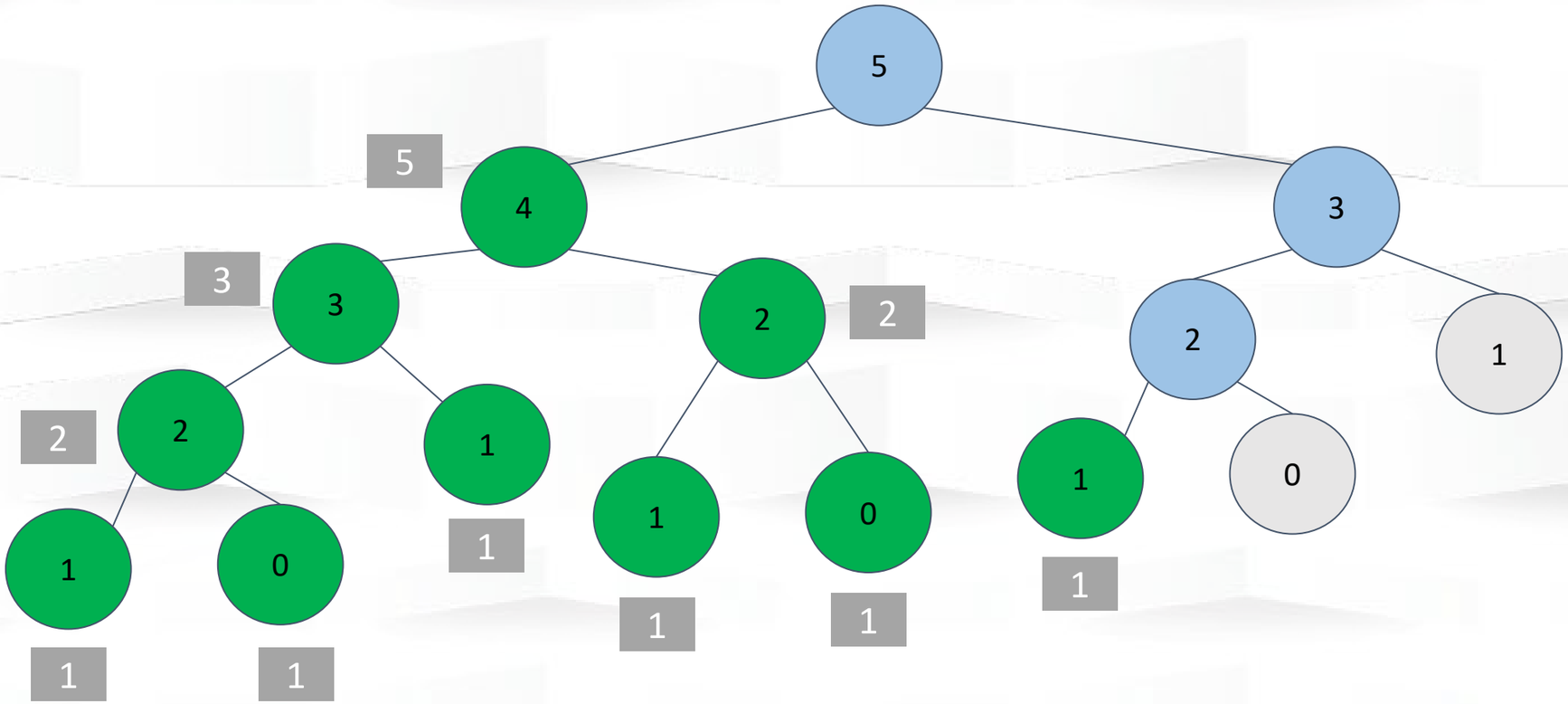
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

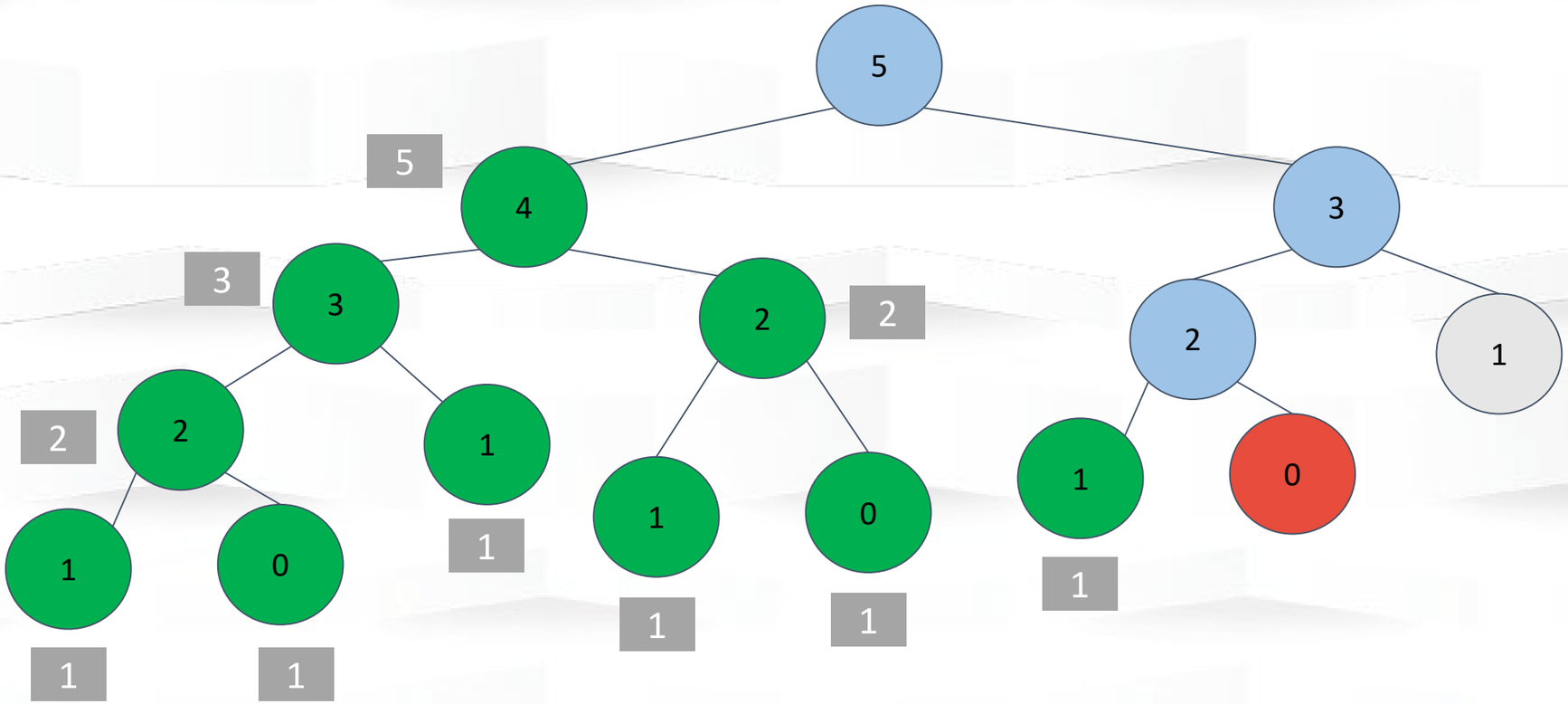
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

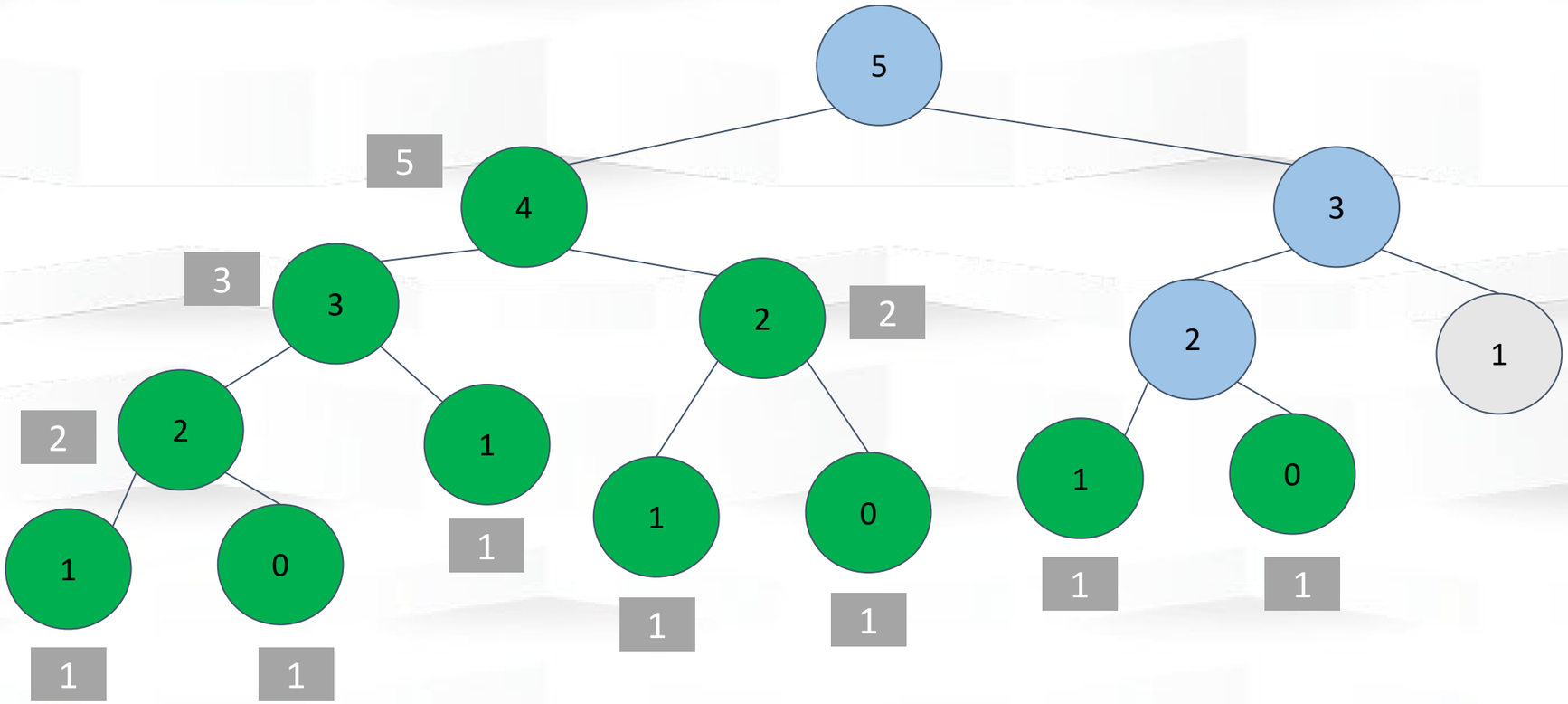
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

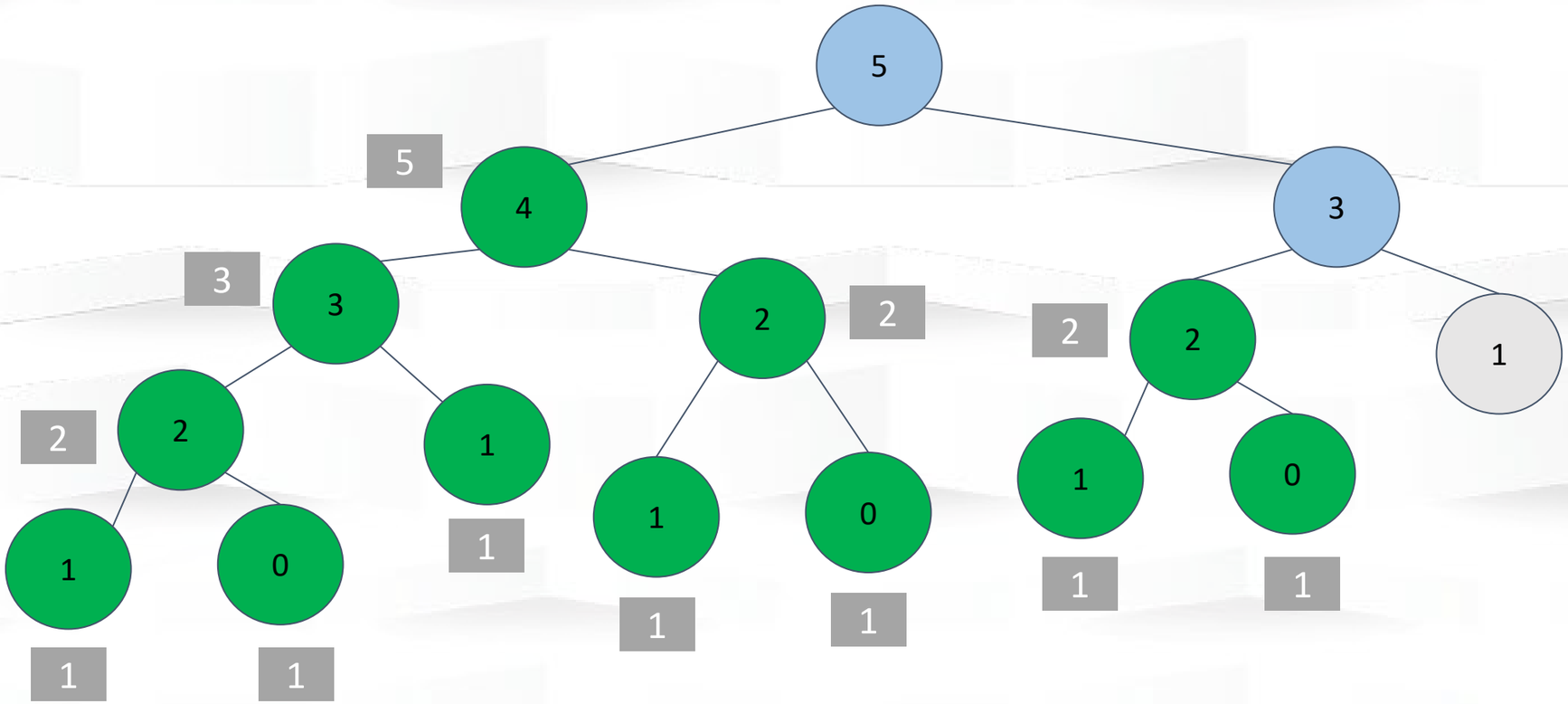
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

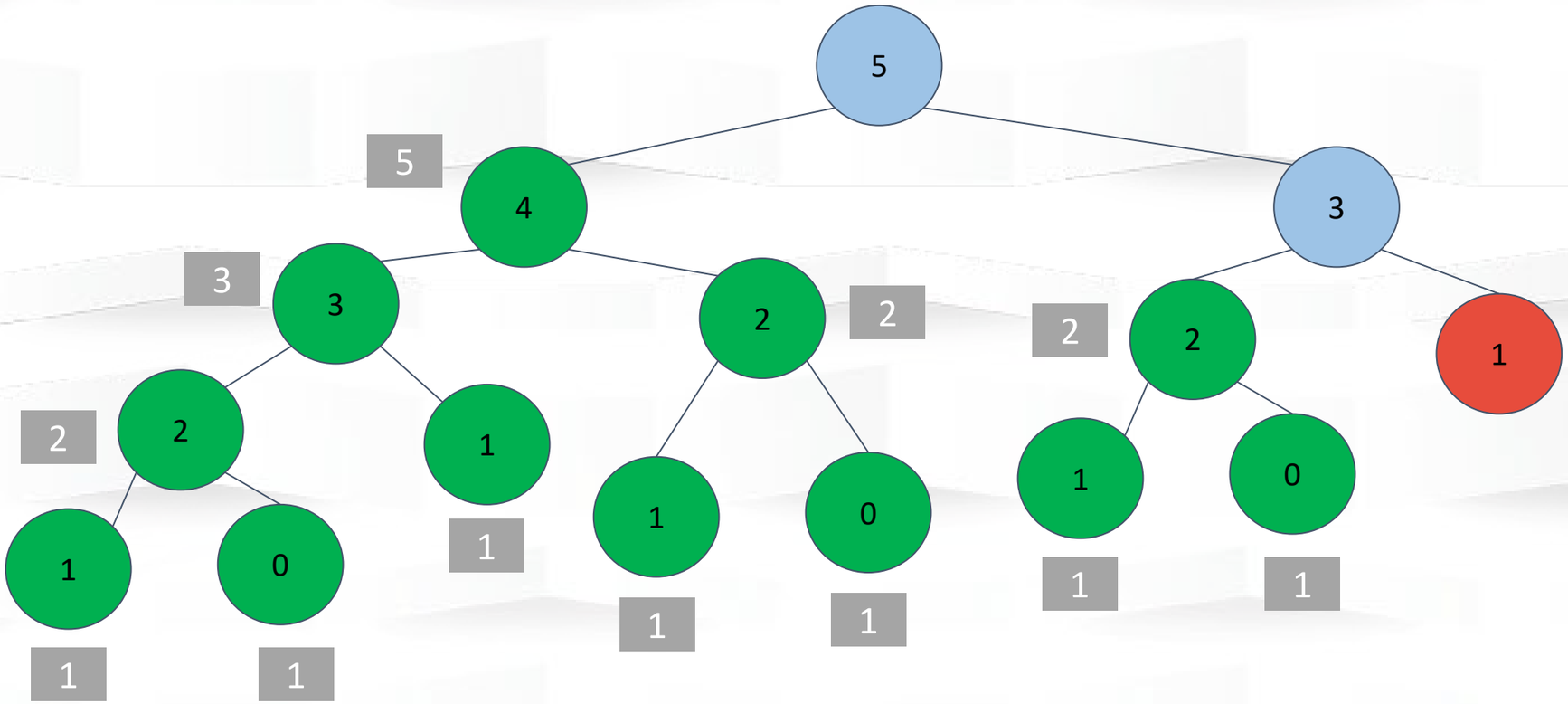
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

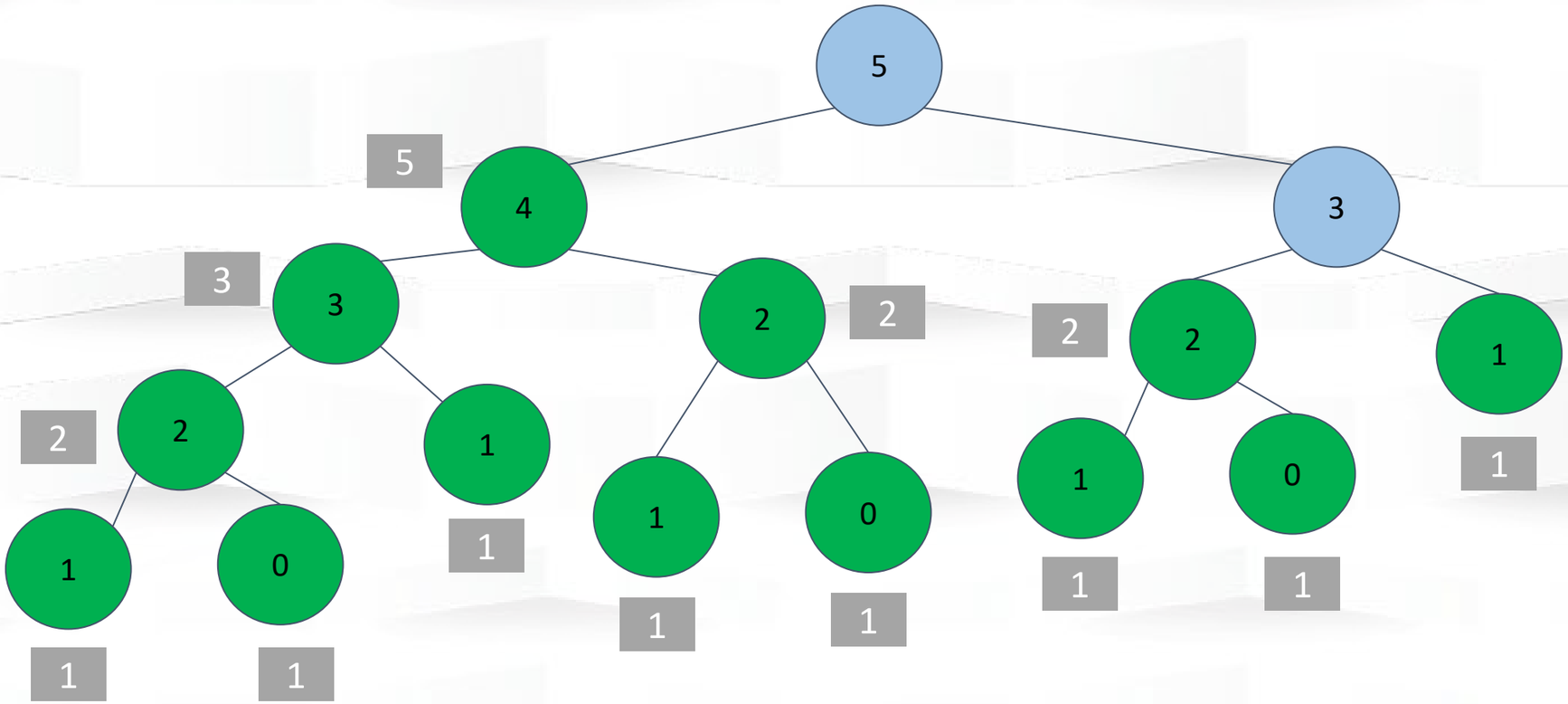
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

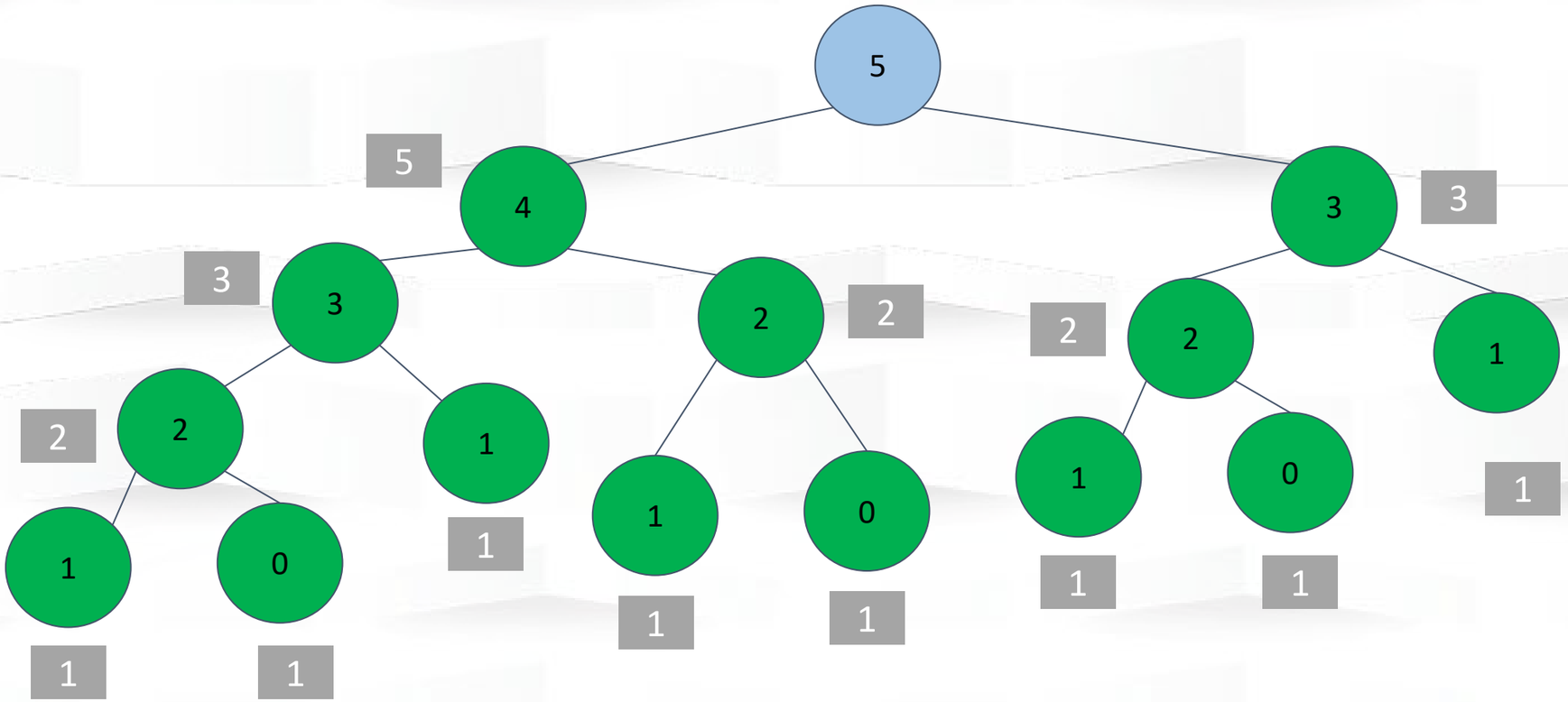
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

Ejemplo: Fibonacci(5)

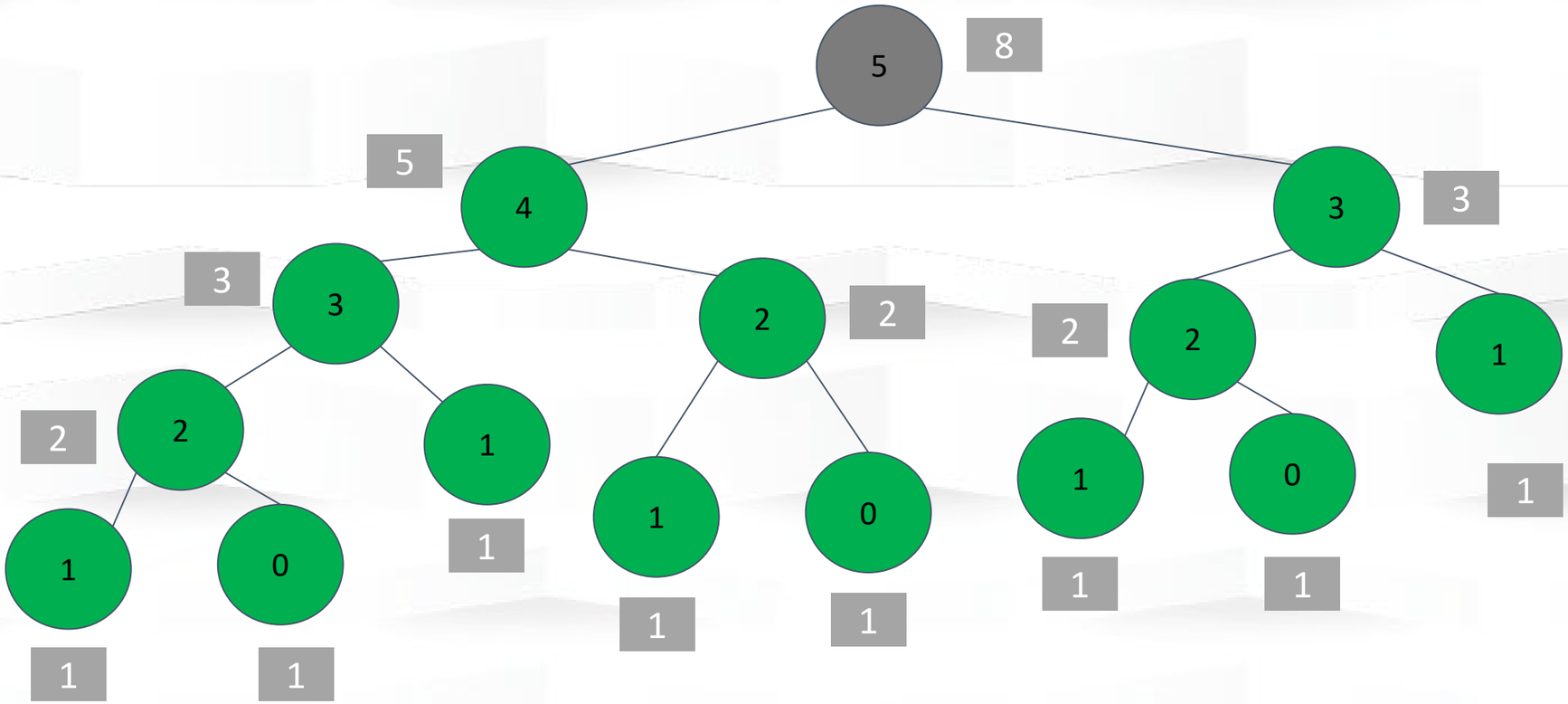


Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

Ejemplo: Fibonacci(5)

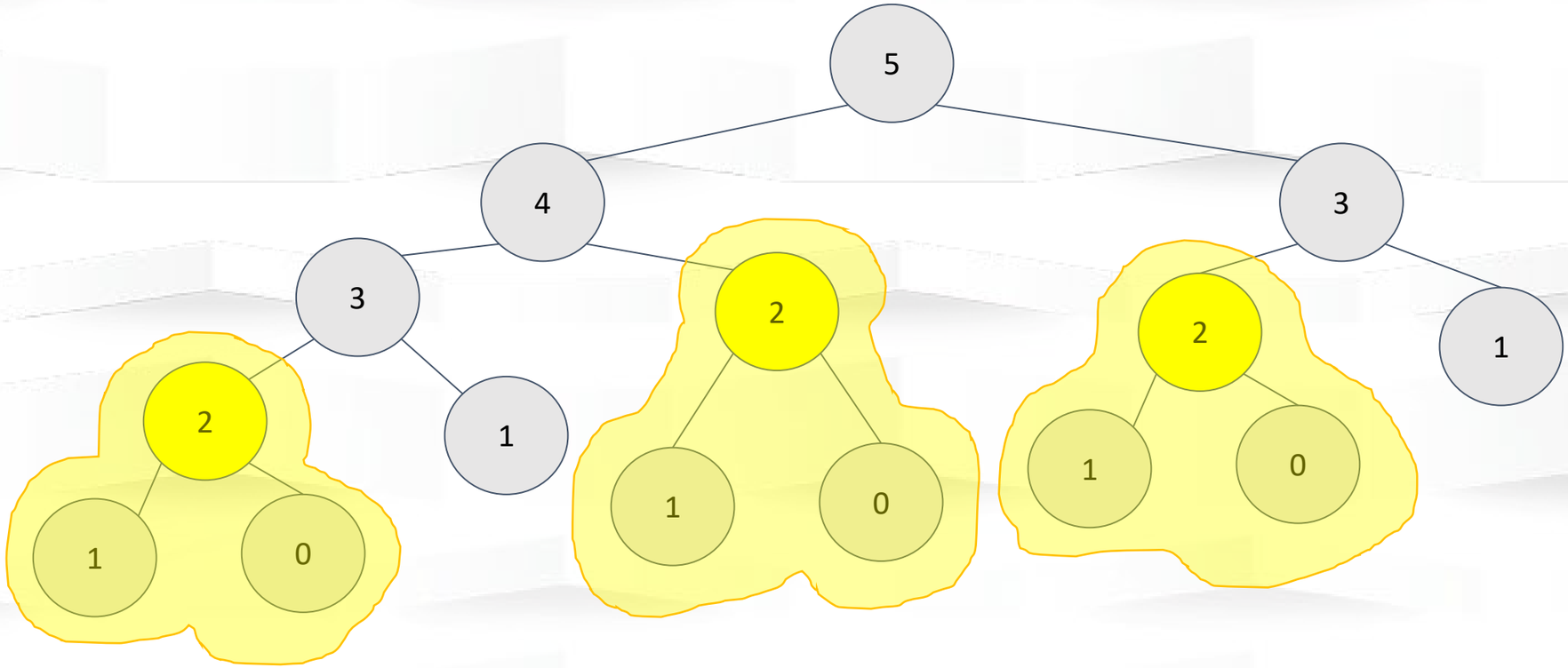
Complejidad **exponencial**



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

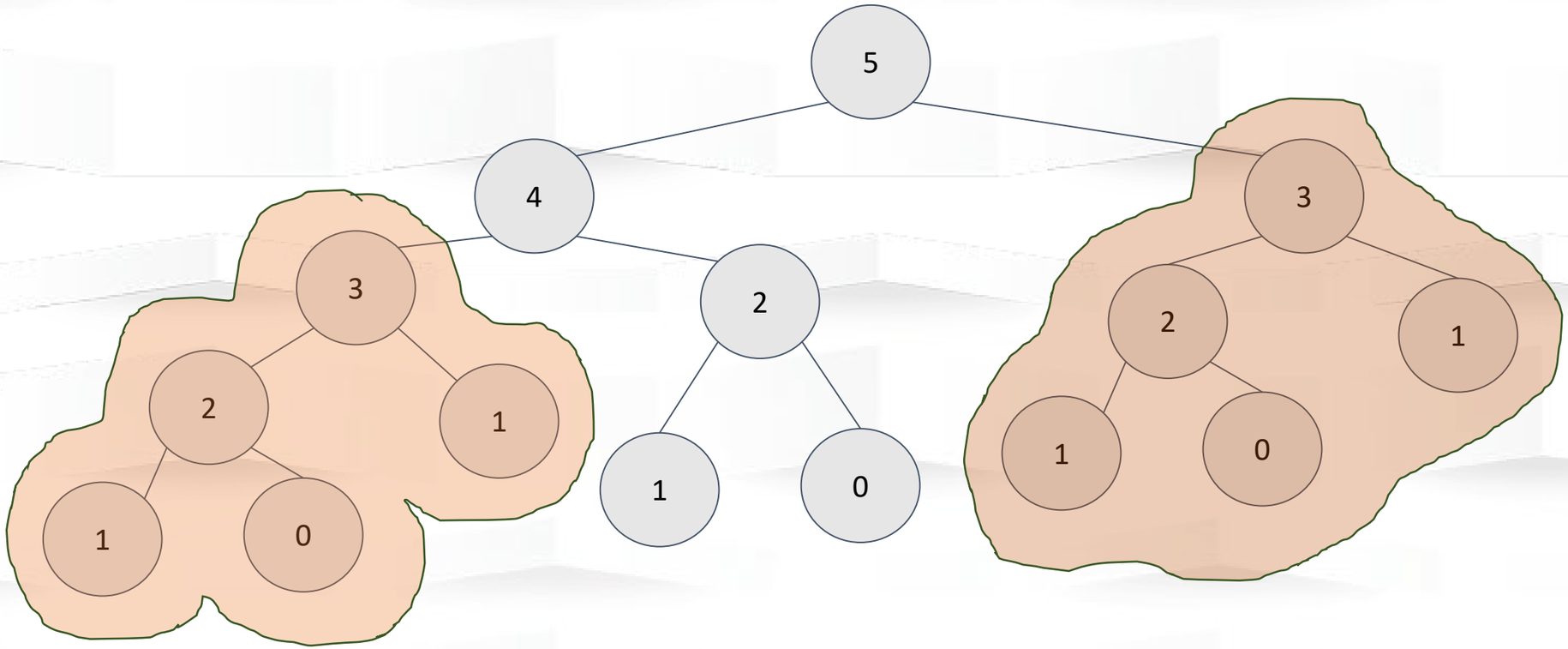
Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

Ejemplo: Fibonacci(5)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

1. Estado
2. Transición
3. Memoria
4. Casos base

Problemas clásicos: maximizar, minimizar, contar caminos, ...

20% de los problemas de un concurso son **DP**

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

1. Estado
   - Situación del problema en la que te encuentras (y que se puede repetir)
   - Está determinado por un conjunto de parámetros
   - Ejemplo (Fibonacci): número de Fibonacci a calcular

2. Transición
3. Memoria
4. Casos base

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

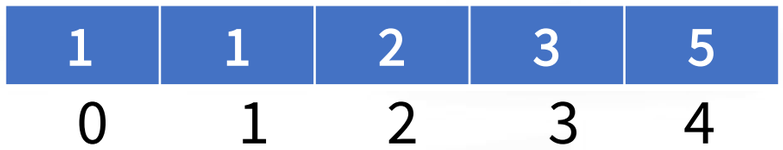
1. Estado
2. Transición
   - Paso de un estado a otro
   - Ejemplo (Fibonacci):
     Fibo(n) **=** Fibo(n-1) + Fibo(n-2)

3. Memoria
4. Casos base

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

1. Estado
2. Transición
3. Memoria
   - Estructura para almacenar los resultados previamente calculados
   - Se accede a ella a través de los parámetros que caracterizan el estado
   - Ejemplo (Fibonacci):



4. Casos base

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

1. Estado
2. Transición
3. Memoria
4. Casos base
   - Estados de los cuales conocemos la solución de antemano
   - Ejemplo (Fibonacci): Fibonacci(0) = 1
     Fibonacci(1) = 1

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

Determinar las características del problema:

- Las características (parámetros) que determinan. totalmente un estado.
- Los casos bases para los cuales conocemos la solución.
- Las transiciones entre estados.
- La estructura de memorización necesaria.

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

O (nº estados * nº transiciones)

Ejemplo Fibonacci: O (n * 2) = **O(n)**

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica - Top Down

- Llegar desde el problema general hasta los casos base.
- Enfoque **RECURSIVO.**
- DP recursivo = Recursión con Fuerza Bruta + Memo.

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica - Top Down

Esquema general de la recursión

```
resolver ( parámetros ) {
   - caso base? -> solución ya conocida
   - en otro caso:
        - transiciones hacia otros estados
}
```

Programación dinámica

```
resolver (parámetros ) {
   - caso base? -> solución ya conocida
   - resultado ya calculado? -> Memo
   - en otro caso:
        - transiciones hacia otros estados
        - almacenar en la Memo y devolver el
                resultado
}
```


Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica - Top Down

Esquema general de la recursión

```
Fibonacci ( n ) {
   si n==0 o n==1: devolver 1;
   si no: devolver Fibonacci(n-1) +
   Fibonacci(n-2)
}
```

Programación dinámica

```
Fibonacci ( n ) {
   si n==0 o n==1: devolver 1;
   si calculado Fibonacci(n): devolver
   Memo(n)
   si no: calcular Fibonacci(n-1) +
   Fibonacci(n-2)
            almacenar Memo(n)
            devolver Memo(n)
}
```


Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica - Bottom Up

- Menos intuitivo.
- Enfoque **ITERATIVO.**
- Llegar desde los casos base al problema general.
- ¿Cómo?

1. Rellenar la memo de los casos base.
2. Rellenar los estados que se pueden calcular a partir de los resueltos.
3. Repetir 2 hasta llegar al caso deseado.

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica

- Cualquier problema se **debería** poder resolver de las 2 formas

	Top-Down	Bottom-Up
PROS	- Transformación natural de recursión por Fuerza Bruta - Solo se calculan los resultados necesarios	- Más rápido si hay muchas llamadas recursivas
CONTRAS	- Más lento si hay muchas llamadas recursivas - Posible RTE por Stack Overflow	- Menos intuitivo - Se calculan todos los resultados

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Programación dinámica - Varios

- Longest Increasing Subsequence (LIS)
- Longest Common Subsequence (LCS)
- Coin Problem
- Return choice
- DP con máscara de bits
- Teoría de juegos

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Ejercicios de programación dinámica

Ejercicios de programación dinámica

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

¿Cómo crear un formulario?

CÓMO CREAR UN FORMULARIO

- Plantilla propuesta: https://www.overleaf.com/read/mhcdkrbxmfnv#c1f2b2

- Depende del concurso ponen límites de página (habitualmente 25 páginas en fuente Arial tamaño 10).

Recordad que los concursos son sin internet, esto es lo único que podemos llevar con nosotros.
Os podemos pasar alguno en Python ☺.

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Consejos para los concursos

CONSEJOS PARA LOS CONCURSOS

Pre-submit:

- Write a few simple test cases, if sample is not enough.
- Are time limits close? If so, generate max cases.
- Is the memory usage fine?
- Could anything overflow?
- Make sure to submit the right file.

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

CONSEJOS PARA LOS CONCURSOS

Wrong answer:

- Print your solution! Print debug output, as well.
- Are you clearing all datastructures between test cases?
- Can your algorithm handle the whole range of input?
- Read the full problem statement again.
- Do you handle all corner cases correctly?
- Have you understood the problem correctly?
- Wrong copy code?

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

CONSEJOS PARA LOS CONCURSOS

Wrong answer:

- Any uninitialized variables?
- Same variable name?
- Correct recursion?
- Confusing N and M, i and j, etc?
- Are you sure your algorithm works?
- What special cases have you not thought of?
- Are you sure the STL functions you use work as you think?

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

CONSEJOS PARA LOS CONCURSOS

Wrong answer:

- Add some assertions, maybe resubmit.
- Create some testcases to run your algorithm on.
- Go through the algorithm for a simple case.
- Go through this list again.
- Explain your algorithm to a team mate.
- Ask the team mate to look at your code.
- Go for a small walk, e.g. to the toilet.
- Is your output format correct? (including whitespace).
- Rewrite your solution from the start.

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

CONSEJOS PARA LOS CONCURSOS

Runtime error:

- Have you tested all corner cases locally?
- Any uninitialized variables?
- Are you reading or writing outside the range of any vector?
- Any assertions that might fail?
- Any possible division by 0? (mod 0 for example).
- Any possible infinite recursion?
- Invalidated pointers or iterators?
- Are you using too much memory?
- Debug with resubmits (e.g. remapped signals, see Various).

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

CONSEJOS PARA LOS CONCURSOS

Time limit exceeded:

- Do you have any possible infinite loops?
- What is the complexity of your algorithm?
- Are you copying a lot of unnecessary data? (References).
- How big is the input and output? (consider scanf)
- Avoid vector, map. (use arrays/unordered_map)
- Use vector? Change to array.
- What do your team mates think about your algorithm?

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

CONSEJOS PARA LOS CONCURSOS

Memory limit exceeded:

- What is the max amount of memory your algorithm should need?
- Are you clearing all datastructures between test cases?

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

CONSEJOS PARA LOS CONCURSOS

Según la experiencia, los que mejores funcionan:

- Borrar ejercicio y volver a empezar.
- Crear casos de prueba a mano.
- Verificar desbordamientos (int, long, etc).
- Hacer algoritmos para validar nuestro código.
- No ser impulsivos y verificar que el programa funciona correctamente según el formulario.
- Usar entrada rápida.

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

PONEDLOS EN PRÁCTICA

La siguiente sesión será un concurso individual presencial el **26 de marzo, de 15:00 a 18:00** en la Universidad de Cantabria.
https://forms.office.com/e/J8rTgwD7UW

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

SIGUIENTES CONCURSOS

- **AdaByron MultiSede** - 9 de abril de 16:00 a 19:00.
- **AdaByron Nacional** - 4 y 5 de julio de 2025 en la Facultad de Informática de la Universidad Complutense de Madrid.

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

#CátedrasCiber

Módulo IV: Fuerza bruta y programación dinámica

Financiado por la Unión Europea
NextGenerationEU

GOBIERNO DE ESPAÑA
MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA
SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL

Plan de Recuperación, Transformación y Resiliencia

Entidad colaboradora:
incibe_
INSTITUTO NACIONAL DE CIBERSEGURIDAD
UC Universidad de Cantabria