

Counting Binary Trees

Joseph Chuang-Chieh Lin (林莊傑)

Department of Computer Science & Engineering,
National Taiwan Ocean University

Fall 2024



Outline

1 Counting Binary Trees



Outline

1 Counting Binary Trees



Counting Binary Trees

- Consider the following three disparate problems:
 - ① The number of distinct binary trees having n nodes.
 - ② The number of distinct permutations of the numbers from 1 to n obtainable by a [stack](#).
 - ③ The number of distinct ways of multiplying $n + 1$ matrices.



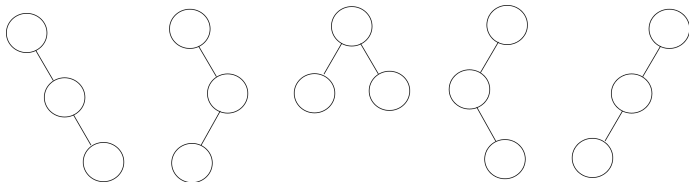
Counting Binary Trees

- Consider the following three disparate problems:
 - ① The number of distinct binary trees having n nodes.
 - ② The number of distinct permutations of the numbers from 1 to n obtainable by a [stack](#).
 - ③ The number of distinct ways of multiplying $n + 1$ matrices.
- Amazingly, **these problems have the same solution!**



Problem One

- The number of distinct binary trees having n nodes.



★ Example of $n = 3$.

Problem Two

- The number of distinct permutations of the numbers from 1 to n obtainable by a **stack**.

- ① push 1 \rightarrow pop \rightarrow push 2 \rightarrow pop \rightarrow push 3 \rightarrow pop \Rightarrow 123.
- ② push 1 \rightarrow pop \rightarrow push 2 \rightarrow push 3 \rightarrow pop \rightarrow pop \Rightarrow 132.
- ③ push 1 \rightarrow push 2 \rightarrow push 3 \rightarrow pop \rightarrow pop \rightarrow pop \Rightarrow 321.
- ④ push 1 \rightarrow push 2 \rightarrow pop \rightarrow pop \rightarrow push 3 \rightarrow pop \Rightarrow 213.
- ⑤ push 1 \rightarrow push 2 \rightarrow pop \rightarrow push 3 \rightarrow pop \rightarrow pop \Rightarrow 231.

★ Example of $n = 3$.



Problem Three

- The number of distinct ways of multiplying $n + 1$ matrices.

① $((M_1 \times M_2) \times M_3) \times M_4.$

② $(M_1 \times (M_2 \times M_3)) \times M_4.$

③ $M_1 \times ((M_2 \times M_3) \times M_4).$

④ $M_1 \times (M_2 \times (M_3 \times M_4)).$

⑤ $(M_1 \times M_2) \times (M_3 \times M_4).$

★ Example of $n = 3$.



Stack Permutation (1/4)

- Recall: preorder, inorder and postorder traversal of a binary tree.
 - Each traversal requires a **stack**.

Every binary tree has a unique pair of preorder/inorder sequences.



Stack Permutation (1/4)

- Recall: preorder, inorder and postorder traversal of a binary tree.
 - Each traversal requires a **stack**.

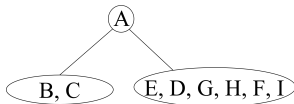
Every binary tree has a unique pair of preorder/inorder sequences.

- The number of distinct binary trees is equal to the number of **inorder permutations** obtainable from binary trees having the preorder permutation, $1, 2, \dots, n$.

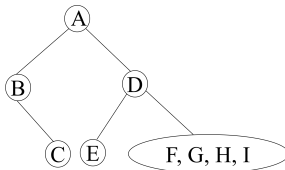


Stack Permutation (2/4)

- preorder: A B C E D G H F I
- inorder: B C A E D G H F I



- preorder: A B C (D E F G H I)
- inorder: B C A (E D F G H I)



Stack Permutation (3/4)

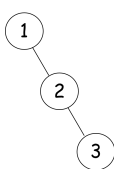
- We can show that

the number of distinct permutations obtainable by passing the numbers $\{1, 2, \dots, n\}$ through a stack is equal to the number of distinct binary trees with n nodes.

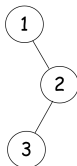
- ① push 1 \rightarrow pop \rightarrow push 2 \rightarrow pop \rightarrow push 3 \rightarrow pop \Rightarrow 123.
- ② push 1 \rightarrow pop \rightarrow push 2 \rightarrow push 3 \rightarrow pop \rightarrow pop \Rightarrow 132.
- ③ push 1 \rightarrow push 2 \rightarrow push 3 \rightarrow pop \rightarrow pop \rightarrow pop \Rightarrow 321.
- ④ push 1 \rightarrow push 2 \rightarrow pop \rightarrow pop \rightarrow push 3 \rightarrow pop \Rightarrow 213.
- ⑤ push 1 \rightarrow push 2 \rightarrow pop \rightarrow push 3 \rightarrow pop \rightarrow pop \Rightarrow 231.



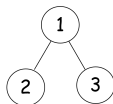
Stack Permutation (4/4)



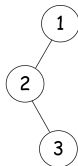
(1, 2, 3)



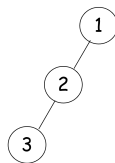
(1, 3, 2)



(2, 1, 3)



(2, 3, 1)



(3, 2, 1)

Go Back to the Matrix Multiplication

- Computing the product of n matrices are related to the distinct binary tree problem.
- $n = 3$:
 - ① $(M_1 \times M_2) \times M_3$.
 - ② $M_1 \times (M_2 \times M_3)$.
- $n = 4$:
 - ① $((M_1 \times M_2) \times M_3) \times M_4$.
 - ② $(M_1 \times (M_2 \times M_3)) \times M_4$.
 - ③ $M_1 \times ((M_2 \times M_3) \times M_4)$.
 - ④ $M_1 \times (M_2 \times (M_3 \times M_4))$.
 - ⑤ $(M_1 \times M_2) \times (M_3 \times M_4)$.



Matrix Multiplication (2/2)

- b_n : the number of different ways to compute the product of n matrices.
- Trivially, $b_1 =$



Matrix Multiplication (2/2)

- b_n : the number of different ways to compute the product of n matrices.
- Trivially, $b_1 = 1$, $b_2 =$



Matrix Multiplication (2/2)

- b_n : the number of different ways to compute the product of n matrices.
- Trivially, $b_1 = 1$, $b_2 = 1$.
- We have also derived that $b_3 = 2$ and $b_4 = 5$.
- We can compute that

$$b_n = \sum_{i=1}^{n-1} b_i b_{n-i}, \text{ for } n > 1.$$



Distinct Binary Trees (1/3)

- Similarly, the number of **distinct binary trees** of n nodes is

$$b_n =$$



Distinct Binary Trees (1/3)

- Similarly, the number of **distinct binary trees** of n nodes is

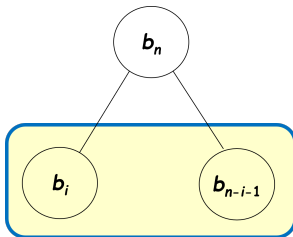
$$b_n = \sum_{i=0}^{n-1} b_i b_{n-1-i}, \text{ for } n \geq 1 \text{ and}$$



Distinct Binary Trees (1/3)

- Similarly, the number of **distinct binary trees** of n nodes is

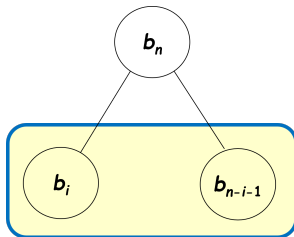
$$b_n = \sum_{i=0}^{n-1} b_i b_{n-1-i}, \text{ for } n \geq 1 \text{ and } b_0 = 1.$$



Distinct Binary Trees (1/3)

- Similarly, the number of **distinct binary trees** of n nodes is

$$b_n = \sum_{i=0}^{n-1} b_i b_{n-1-i}, \text{ for } n \geq 1 \text{ and } b_0 = 1.$$



- But, how to compute b_n exactly?



The Generating Function Trick

- **Trick:** Let $B(x) = \sum_{i \geq 0} b_i x^i$ be the generating function for the number of binary trees.



The Generating Function Trick

- **Trick:** Let $B(x) = \sum_{i \geq 0} b_i x^i$ be the generating function for the number of binary trees.
- By the recurrence relation we get:

$$xB(x)^2 = B(x) - 1.$$

- Solving the recurrence relation, we have

$$\begin{aligned} B(x) &= \frac{1 - \sqrt{1 - 4x}}{2x} \\ &= \frac{1}{2x} \left(1 - \sum_{i \geq 0} \binom{1/2}{i} (-4x)^i \right) \\ &= \sum_{m \geq 0} \binom{1/2}{m+1} (-1)^m 2^{2m+1} x^m. \end{aligned}$$



The Generating Function Trick

- **Trick:** Let $B(x) = \sum_{i \geq 0} b_i x^i$ be the generating function for the number of binary trees.
- By the recurrence relation we get:

$$xB(x)^2 = B(x) - 1.$$

- Solving the recurrence relation, we have

$$\begin{aligned} B(x) &= \frac{1 - \sqrt{1 - 4x}}{2x} & \therefore b_n &= \frac{1}{n+1} \binom{2n}{n}. \\ &= \frac{1}{2x} \left(1 - \sum_{i \geq 0} \binom{1/2}{i} (-4x)^i \right) \\ &= \sum_{m \geq 0} \binom{1/2}{m+1} (-1)^m 2^{2m+1} x^m. \end{aligned}$$



Discussions

