

Counting Binary Trees & Selection Trees

Joseph Chuang-Chieh Lin (林莊傑)

Department of Computer Science & Engineering,
National Taiwan Ocean University

Fall 2025



Outline

- 1 Counting Binary Trees
- 2 Selection Trees

Outline

1 Counting Binary Trees

2 Selection Trees

Counting Binary Trees

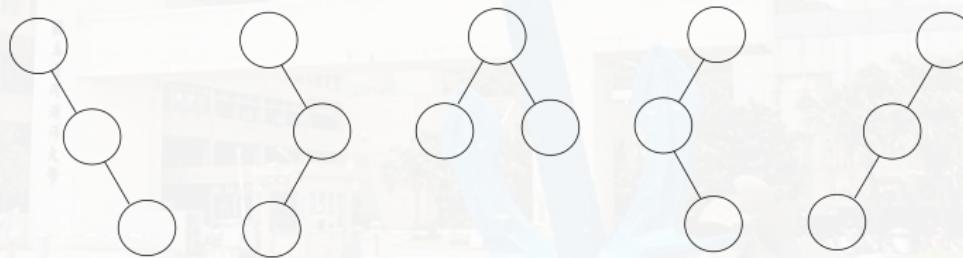
- Consider the following three disparate problems:
 - The number of distinct binary trees having n nodes.
 - The number of distinct permutations of the numbers from 1 to n obtainable by a **stack**.
 - The number of distinct ways of multiplying $n + 1$ matrices.

Counting Binary Trees

- Consider the following three disparate problems:
 - ➊ The number of distinct binary trees having n nodes.
 - ➋ The number of distinct permutations of the numbers from 1 to n obtainable by a **stack**.
 - ➌ 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 → pop → push 2 → pop → push 3 → pop ⇒ 123.
 - ② push 1 → pop → push 2 → push 3 → pop → pop ⇒ 132.
 - ③ push 1 → push 2 → push 3 → pop → pop → pop ⇒ 321.
 - ④ push 1 → push 2 → pop → pop → push 3 → pop ⇒ 213.
 - ⑤ push 1 → push 2 → pop → push 3 → pop → pop ⇒ 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)

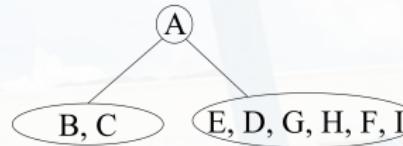
- Recall: preorder, inorder and postorder traversal of a binary tree.
 - Each traversal requires a **stack**.

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

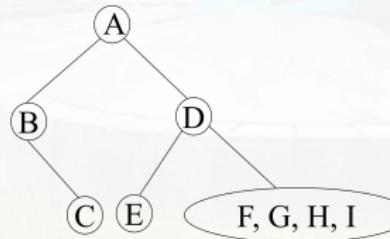
- The number of distinct binary trees is equal to the number of **inorder permutations** obtainable from binary trees having the preorder permutation, $1, 2, \dots, n$.

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

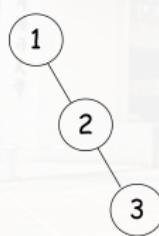
the number of distinct permutations obtainable by passing the numbers $\{1, 2, \dots, n\}$ through a stack is equal to the number of distinct binary trees with n nodes.

- ① push 1 → pop → push 2 → pop → push 3 → pop ⇒ 123.
- ② push 1 → pop → push 2 → push 3 → pop → pop ⇒ 132.
- ③ push 1 → push 2 → push 3 → pop → pop → pop ⇒ 321.
- ④ push 1 → push 2 → pop → pop → push 3 → pop ⇒ 213.
- ⑤ push 1 → push 2 → pop → push 3 → pop → pop ⇒ 231.

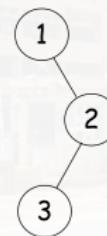


Stack Permutation (4/4)

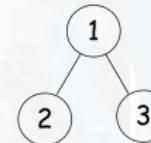
- visit node → push to stack;
 - going left → keep visiting next node
 - going right → pop the stack
- the leaf → pop the stack until empty



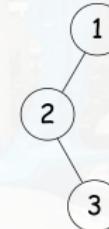
(1, 2, 3)



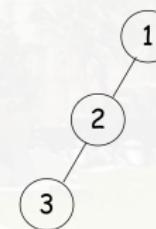
(1, 3, 2)



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

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 $((M_1 \times M_2) \times M_3) \times M_4$ (push, pop, push, pop, push, pop)
 - 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

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

$$b_n =$$

Distinct Binary Trees

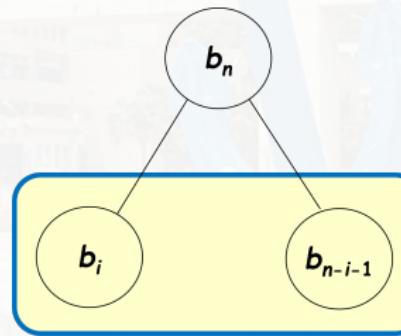
- Similarly, the number of **distinct binary trees** of n nodes is

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

Distinct Binary Trees

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

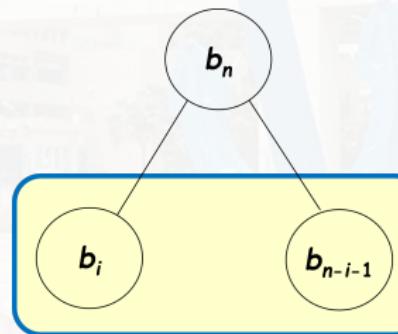
$$b_n = \sum_{i=0}^{n-1} b_i b_{n-1-i}, \text{ for } n \geq 1 \text{ and } b_0 = 1.$$



Distinct Binary Trees

- 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.

$$B(x) = \sum_{i \geq 0} b_i x^i = \sum_{i=0}^{\infty} \sum_{k=0}^{i-1} b_k b_{i-1-k} x^i =$$

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.

$$\begin{aligned} B(x) &= \sum_{i \geq 0} b_i x^i = \sum_{i=0}^{\infty} \sum_{k=0}^{i-1} b_k b_{i-1-k} x^i &= 1 + \sum_{i=1}^{\infty} \sum_{k=0}^{i-1} b_k b_{i-1-k} x^i \\ &= \end{aligned}$$



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.

$$\begin{aligned}
 B(x) &= \sum_{i \geq 0} b_i x^i = \sum_{i=0}^{\infty} \sum_{k=0}^{i-1} b_k b_{i-1-k} x^i &= 1 + \sum_{i=1}^{\infty} \sum_{k=0}^{i-1} b_k b_{i-1-k} x^i \\
 &= 1 + \cancel{x} \sum_{i=1}^{\infty} \sum_{k=0}^{i-1} b_k b_{i-1-k} x^{i-1} \\
 &=
 \end{aligned}$$

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.

$$\begin{aligned}
 B(x) &= \sum_{i \geq 0} b_i x^i = \sum_{i=0}^{\infty} \sum_{k=0}^{i-1} b_k b_{i-1-k} x^i = 1 + \sum_{i=1}^{\infty} \sum_{k=0}^{i-1} b_k b_{i-1-k} x^i \\
 &= 1 + \cancel{x} \sum_{i=1}^{\infty} \sum_{k=0}^{i-1} b_k b_{i-1-k} x^{i-1} \\
 &= 1 + x \sum_{j=0}^{\infty} \sum_{k=0}^j b_k b_{j-k} x^j \\
 &= 1 + x \left(\sum_{j=0}^{\infty} b_j x^j \right)^2 = 1 + x B(x)^2.
 \end{aligned}$$



The Generating Function Trick

- By the recurrence relation we get:

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

- Solving the recurrence relation, we have

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

The Generating Function Trick

- By the recurrence relation we get:

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

- Solving the recurrence relation, we have

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

By the Binomial Theorem...

$$b_n = \binom{1/2}{n+1} (-1)^n 2^{2n+1}$$

=

By the Binomial Theorem...

$$\begin{aligned} b_n &= \binom{1/2}{n+1} (-1)^n 2^{2n+1} \\ &= \frac{\frac{1}{2} \left(\frac{1}{2}-1\right) \left(\frac{1}{2}-2\right) \cdots \left(\frac{1}{2}-n\right)}{(n+1)n(n-1)(n-2)\cdots 2 \cdot 1} (-1)^n 2^{2n+1} \\ &= \end{aligned}$$

By the Binomial Theorem...

$$\begin{aligned} b_n &= \binom{1/2}{n+1} (-1)^n 2^{2n+1} \\ &= \frac{\frac{1}{2} \left(\frac{1}{2}-1\right) \left(\frac{1}{2}-2\right) \cdots \left(\frac{1}{2}-n\right)}{(n+1)n(n-1)(n-2)\cdots 2 \cdot 1} (-1)^n 2^{2n+1} \\ &= \frac{\frac{1}{2} \left(1-\frac{1}{2}\right) \left(2-\frac{1}{2}\right) \cdots \left(n-\frac{1}{2}\right)}{(n+1)n(n-1)(n-2)\cdots 2 \cdot 1} 2^{2n+1} \\ &= \end{aligned}$$

By the Binomial Theorem...

$$\begin{aligned} b_n &= \binom{1/2}{n+1} (-1)^n 2^{2n+1} \\ &= \frac{\frac{1}{2} \left(\frac{1}{2}-1\right) \left(\frac{1}{2}-2\right) \cdots \left(\frac{1}{2}-n\right)}{(n+1)n(n-1)(n-2)\cdots 2 \cdot 1} (-1)^n 2^{2n+1} \\ &= \frac{\frac{1}{2} \left(1-\frac{1}{2}\right) \left(2-\frac{1}{2}\right) \cdots \left(n-\frac{1}{2}\right)}{(n+1)n(n-1)(n-2)\cdots 2 \cdot 1} 2^{2n+1} \\ &= \left(\frac{1}{2}\right)^{n+1} \cdot \frac{1(2-1)(4-1)\cdots(2n-1)}{(n+1)n(n-1)(n-2)\cdots 2 \cdot 1} 2^{2n+1} \\ &= \end{aligned}$$

By the Binomial Theorem...

$$\begin{aligned}
 b_n &= \binom{1/2}{n+1} (-1)^n 2^{2n+1} \\
 &= \frac{\frac{1}{2} \left(\frac{1}{2}-1\right) \left(\frac{1}{2}-2\right) \cdots \left(\frac{1}{2}-n\right)}{(n+1)n(n-1)(n-2)\cdots 2 \cdot 1} (-1)^n 2^{2n+1} \\
 &= \frac{\frac{1}{2} \left(1-\frac{1}{2}\right) \left(2-\frac{1}{2}\right) \cdots \left(n-\frac{1}{2}\right)}{(n+1)n(n-1)(n-2)\cdots 2 \cdot 1} 2^{2n+1} \\
 &= \left(\frac{1}{2}\right)^{n+1} \cdot \frac{1(2-1)(4-1)\cdots(2n-1)}{(n+1)n(n-1)(n-2)\cdots 2 \cdot 1} 2^{2n+1} \\
 &= \frac{1(2-1)(4-1)\cdots(2n-1)}{(n+1)n(n-1)(n-2)\cdots 2 \cdot 1} \cdot 2^n \\
 &=
 \end{aligned}$$

By the Binomial Theorem...

$$\begin{aligned}
b_n &= \binom{1/2}{n+1} (-1)^n 2^{2n+1} \\
&= \frac{\frac{1}{2} \left(\frac{1}{2}-1\right) \left(\frac{1}{2}-2\right) \cdots \left(\frac{1}{2}-n\right)}{(n+1)n(n-1)(n-2)\cdots 2 \cdot 1} (-1)^n 2^{2n+1} \\
&= \frac{\frac{1}{2} \left(1-\frac{1}{2}\right) \left(2-\frac{1}{2}\right) \cdots \left(n-\frac{1}{2}\right)}{(n+1)n(n-1)(n-2)\cdots 2 \cdot 1} 2^{2n+1} \\
&= \left(\frac{1}{2}\right)^{n+1} \cdot \frac{1(2-1)(4-1)\cdots(2n-1)}{(n+1)n(n-1)(n-2)\cdots 2 \cdot 1} 2^{2n+1} \\
&= \frac{1(2-1)(4-1)\cdots(2n-1)}{(n+1)n(n-1)(n-2)\cdots 2 \cdot 1} \cdot 2^n \\
&= \frac{1(1)(3)\cdots(2n-1)}{(n+1)n(n-1)(n-2)\cdots 2 \cdot 1} \cdot 2^n \cdot \frac{2 \cdot 4 \cdot 6 \cdots (2n)}{2^n(1 \cdot 2 \cdots n)} \\
&\equiv
\end{aligned}$$



By the Binomial Theorem...

$$\begin{aligned}
 b_n &= \binom{1/2}{n+1} (-1)^n 2^{2n+1} \\
 &= \frac{\frac{1}{2} \left(\frac{1}{2}-1\right) \left(\frac{1}{2}-2\right) \cdots \left(\frac{1}{2}-n\right)}{(n+1)n(n-1)(n-2)\cdots 2\cdot 1} (-1)^n 2^{2n+1} \\
 &= \frac{\frac{1}{2} \left(1-\frac{1}{2}\right) \left(2-\frac{1}{2}\right) \cdots \left(n-\frac{1}{2}\right)}{(n+1)n(n-1)(n-2)\cdots 2\cdot 1} 2^{2n+1} \\
 &= \left(\frac{1}{2}\right)^{n+1} \cdot \frac{1(2-1)(4-1)\cdots(2n-1)}{(n+1)n(n-1)(n-2)\cdots 2\cdot 1} 2^{2n+1} \\
 &= \frac{1(2-1)(4-1)\cdots(2n-1)}{(n+1)n(n-1)(n-2)\cdots 2\cdot 1} \cdot 2^n \\
 &= \frac{1(1)(3)\cdots(2n-1)}{(n+1)n(n-1)(n-2)\cdots 2\cdot 1} \cdot 2^n \cdot \frac{2\cdot 4\cdot 6\cdots(2n)}{2^n(1\cdot 2\cdots n)} \\
 &= \frac{(2n)!}{(n+1)n!n!}
 \end{aligned}$$

By the Binomial Theorem...

$$\begin{aligned}
 b_n &= \binom{1/2}{n+1} (-1)^n 2^{2n+1} \\
 &= \frac{\frac{1}{2} \left(\frac{1}{2}-1\right) \left(\frac{1}{2}-2\right) \cdots \left(\frac{1}{2}-n\right)}{(n+1)n(n-1)(n-2)\cdots 2\cdot 1} (-1)^n 2^{2n+1} \\
 &= \frac{\frac{1}{2} \left(1-\frac{1}{2}\right) \left(2-\frac{1}{2}\right) \cdots \left(n-\frac{1}{2}\right)}{(n+1)n(n-1)(n-2)\cdots 2\cdot 1} 2^{2n+1} \\
 &= \left(\frac{1}{2}\right)^{n+1} \cdot \frac{1(2-1)(4-1)\cdots(2n-1)}{(n+1)n(n-1)(n-2)\cdots 2\cdot 1} 2^{2n+1} \\
 &= \frac{1(2-1)(4-1)\cdots(2n-1)}{(n+1)n(n-1)(n-2)\cdots 2\cdot 1} \cdot 2^n \\
 &= \frac{1(1)(3)\cdots(2n-1)}{(n+1)n(n-1)(n-2)\cdots 2\cdot 1} \cdot 2^n \cdot \frac{2\cdot 4\cdot 6\cdots(2n)}{2^n(1\cdot 2\cdots n)} \\
 &= \frac{(2n)!}{(n+1)n!n!} = \frac{1}{n+1} \binom{2n}{n}.
 \end{aligned}$$

By the Binomial Theorem...

$$\begin{aligned}
 b_n &= \binom{1/2}{n+1} (-1)^n 2^{2n+1} \\
 &= \frac{\frac{1}{2} \left(\frac{1}{2}-1\right) \left(\frac{1}{2}-2\right) \cdots \left(\frac{1}{2}-n\right)}{(n+1)n(n-1)(n-2)\cdots 2\cdot 1} (-1)^n 2^{2n+1} \\
 &= \frac{\frac{1}{2} \left(1-\frac{1}{2}\right) \left(2-\frac{1}{2}\right) \cdots \left(n-\frac{1}{2}\right)}{(n+1)n(n-1)(n-2)\cdots 2\cdot 1} 2^{2n+1} \\
 &= \left(\frac{1}{2}\right)^{n+1} \cdot \frac{1(2-1)(4-1)\cdots(2n-1)}{(n+1)n(n-1)(n-2)\cdots 2\cdot 1} 2^{2n+1} \\
 &= \frac{1(2-1)(4-1)\cdots(2n-1)}{(n+1)n(n-1)(n-2)\cdots 2\cdot 1} \cdot 2^n \\
 &= \frac{1(1)(3)\cdots(2n-1)}{(n+1)n(n-1)(n-2)\cdots 2\cdot 1} \cdot 2^n \cdot \frac{2\cdot 4\cdot 6\cdots(2n)}{2^n(1\cdot 2\cdots n)} \\
 &= \frac{(2n)!}{(n+1)n!n!} = \frac{1}{n+1} \binom{2n}{n}.
 \end{aligned}$$

* supplementary: Stirling's approximation

Outline

1 Counting Binary Trees

2 Selection Trees

Winner Selection Tree

- In the following figure, computing the first winner takes



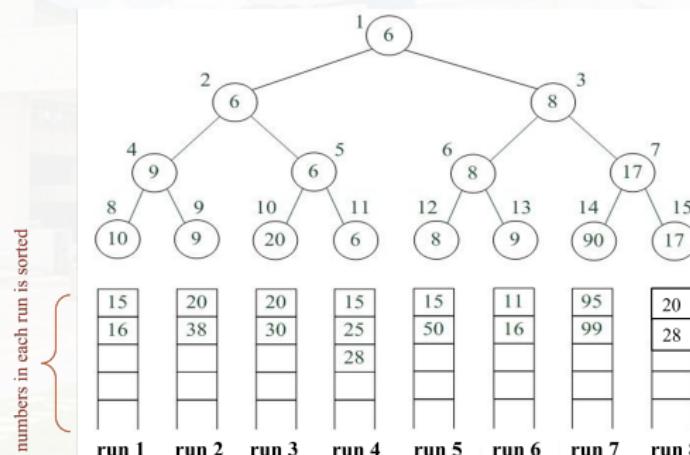
Winner Selection Tree

- In the following figure, computing the first winner takes $k - 1$ comparisons.



Winner Selection Tree

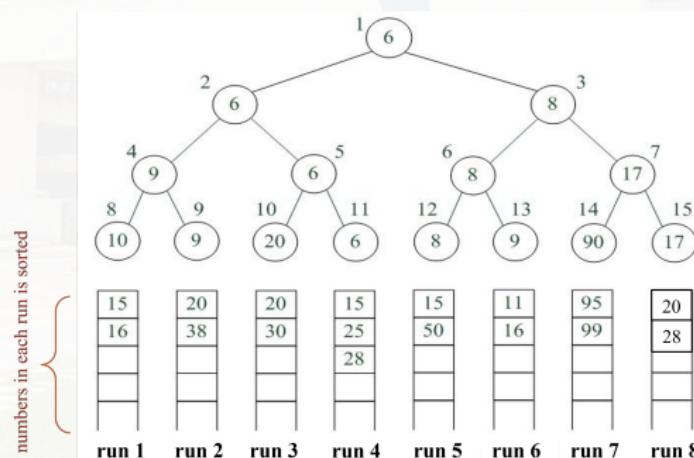
- In the following figure, computing the first winner takes $k - 1$ comparisons.
(not better?)
- But wait, how about the following iterations?



Winner Selection Tree

(Winner) Selection Tree:

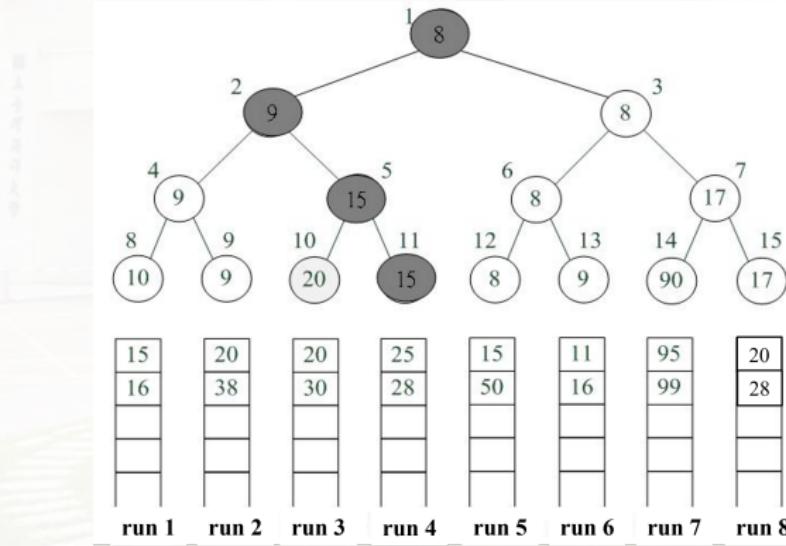
- Setting up the selection tree: $O(k)$ time.
- Restructuring: $O(\lg k)$ time.
- merging all n items: $O(n \lg k)$ time.



Winner Selection Tree

(Winner) Selection Tree:

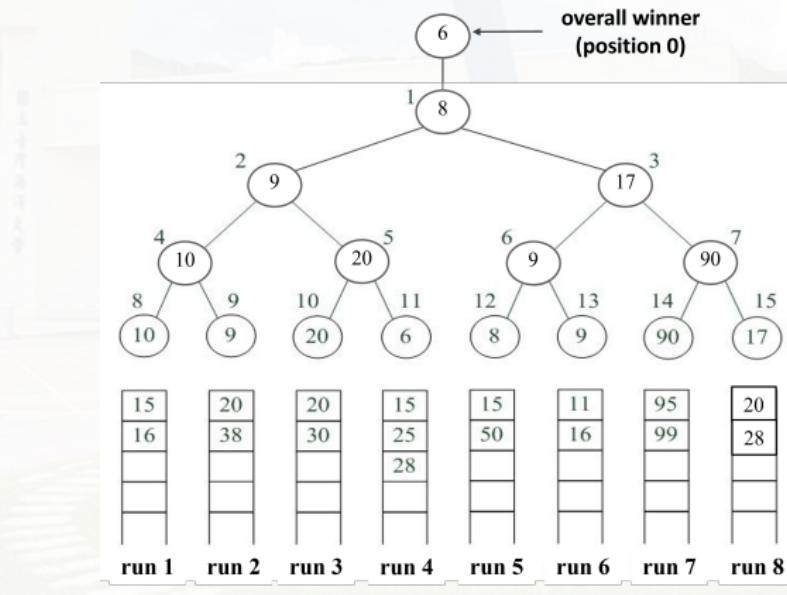
- Setting up the selection tree: $O(k)$ time.
- Restructuring: $O(\lg k)$ time.
- merging all n items: $O(n \lg k)$ time.



Loser Selection Tree

Loser Tree (after the winner tree has been built)

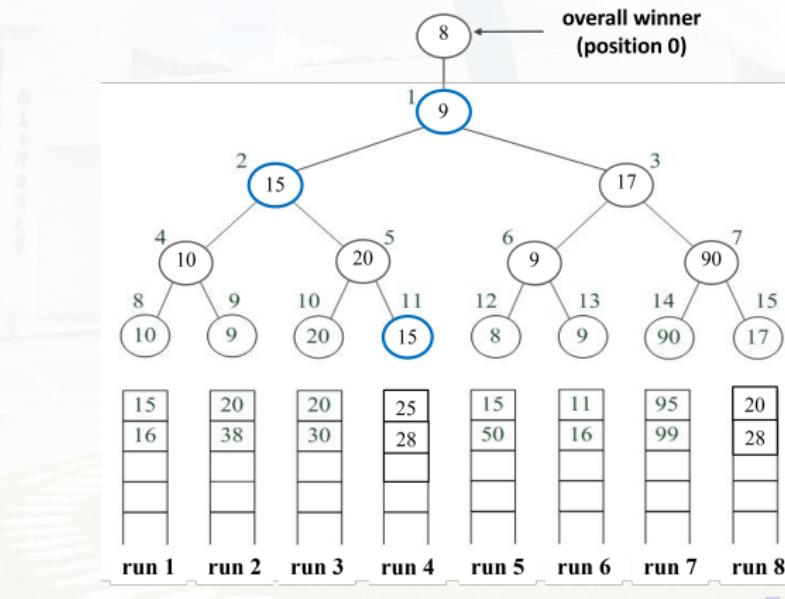
- Sibling nodes represent the losers.
- The restructuring process can be simplified.



Loser Selection Tree

Loser Tree (after the winner tree has been built)

- Sibling nodes represent the losers.
- The restructuring process can be simplified.



Note for the Loser Selection Tree

- Comparison with the sibling is required for the first construction.
- After the first construction, we only need to compare each node with its parent; “push” the smaller key value upward and left the “larger” key value as the **loser**.

Discussions

