

1. Incremental :

Bubble

3 2 1
 2 3 1
 2 1 3
 1 2 3

Selection

3 2 1
 1 3 2
 1 2 3

Insertion

3 2 1
 2 3 1

2. Divide & Conquer

Merge

1 5 7 2 4 6
 ↑ ↑ ↑

1 2 4 5 6 7

Quick

4 5 7 2 3 6
 Δ ↑ ↑
 ↑ ↑

4 3 7 2 5 6
 Δ ↑ ↑

4 3 2 7 5 6
 Δ X

2 3 4 7 5 6
 Δ Δ Δ

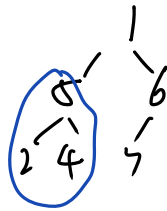
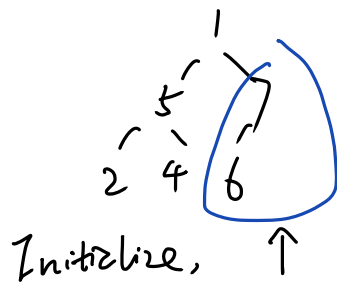
3. Data Structure

heap

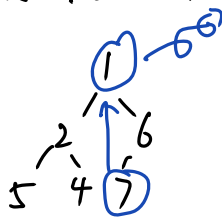
1 5 7 2 4 6

Binary tree

↳ parent < children



Pick min, ↓



tree

BST

↳ parent > left
parent < right



4. Special Assumptions

Counting Sort

V 2 0 2 3 0 1

C

2	1	2	1
---	---	---	---

0 1 2 3

C

2	3	5	6
---	---	---	---

0 1 2 3

V 2 0 2 3 0 1

R

	0	1			3
1	2	3	4	5	6

C

2	3	5	6
---	---	---	---

0 1 2 3

Radix Sort

$$\begin{array}{r|l} 17 & 0 \\ 4 & 5 \\ 9 & 0 \\ 80 & 2 \\ & 9 \\ 6 & 6 \end{array}$$

1	7	0
	9	0
8	0	2
	4	5
	6	6
		9

8	0	2
		9
	4	5
	6	6
	7	7
	9	7

1. Incremental :

Bubble 换
Time: worst $O(n^2)$
Space: $O(1)$
Stable: \checkmark

Selection Pick min
Time: $O(n^2)$
Space: $O(1)$

Insertion
Poker
Time: worst $O(n^2)$
Space: $O(1)$
Stable: \checkmark
Online: \checkmark

Python OrderedDict

2. Divide & Conquer

Merge
Time: $O(n \log n)$
Space: $O(n)$
Stable: \checkmark

\Downarrow

TimSort :

Merge + Insertion

Python sort()

(1) stable \checkmark

(2) Time: $O(n \log n)$

(3) Space: $< O(\ln)$

Quick
Time: $O(n \log n)$
Space: $O(1)$ \downarrow recursion

"In-place"

{ extra space for data
extra space for operations

3. Data Structure

heap

Time: $O(n \log n)$

Space: $O(1)$

Priority queue

tree

Time: $O(n \log n)$

Space: ?

C++ map

* Comparison based. lower bound $O(n \log n)$

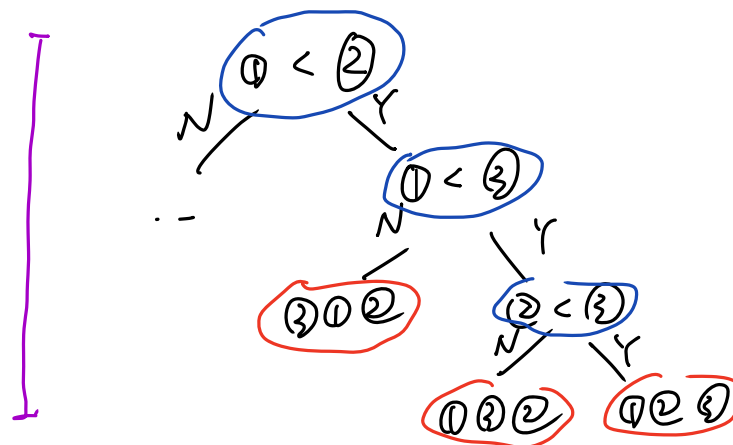
- Decision tree
- Permutation $n!$
- Comparison
- # operations

Leaf

Node

Depth

→ 1 2 3 ($n=3$) 6 pp



$$n! \leq 2^h$$

Goal: $h \geq ?$

Fact: Stirling $n! \geq \left(\frac{n}{e}\right)^n$

$$2^h \geq \left(\frac{n}{e}\right)^n$$

$$h \geq n \log\left(\frac{n}{e}\right) = n \log n - n \log e$$

i.e. $O(n \log n)$

Therefore, the best you can do is $O(n \log n)$.

4. Special Assumptions

Counting Sort

Time: $O(n+k)$

Space: $O(n+k)$

Stable: \checkmark

Assumption: ① integer
② range small

Radix Sort

Time: $O(nk)$

Space: $O(n+k)$

Assumption: ① integer
② range small

* C++ map $O(n \log n)$

Python dict $\approx O(1)$ Hash table

1. Tree \rightarrow Balanced tree
2. Hash table.