Data structures.

Array

- continen , fixed index, can do random Access,
- memory allocted
  - static : fixed → 乙尺osize
  - dynamic
- 排序 complexity ★
  - insert
  - delete
  - lookup
- algorithm
  O(n) , O(nlogn) ...
- 優 : random Access O(1)
- 劣 : 空間利用率(在不確定資料量的情況下)低
  ⇒ hash table (use array) ⇒ if function design badly, there may be lot of space don't be used

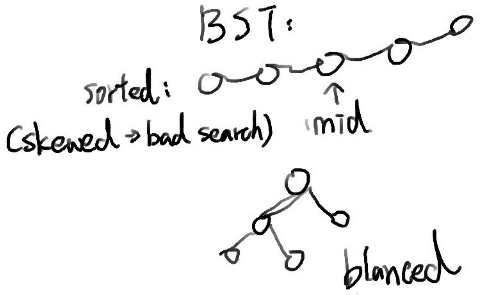
Random Access : O(1)

順序的
sequential search: O(n)

⇒ after sorting : O(logn)

ex - binary search (rule)

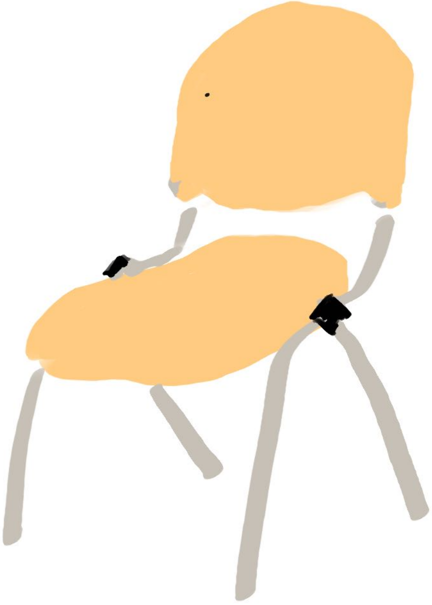
BST:



efficient (adj)

linked list

- Node & pointer
- traverse to find data =O(n)
  Random Access : not exist.
- sequential search : O(n)



Stack/Queue

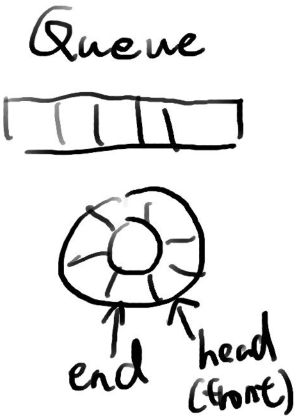
FIFO :

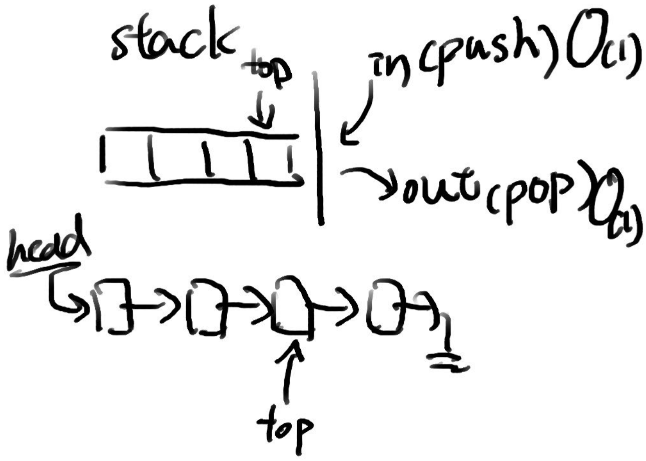
FILO : function call.

implment :

Array

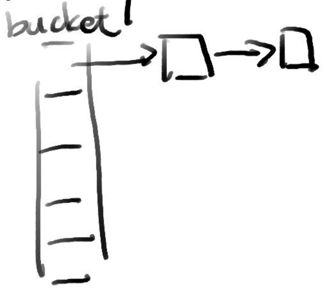
linked list





hash

Array + linked list



→ travers bad (in linked list)

- collision

- Open Addressing no □→ (can use random Aceess)

  - linear probing

    P primary clusterin
    (某個地方特別擠)

  - quadratic probing

    P secondary clustering

  - Double hashing

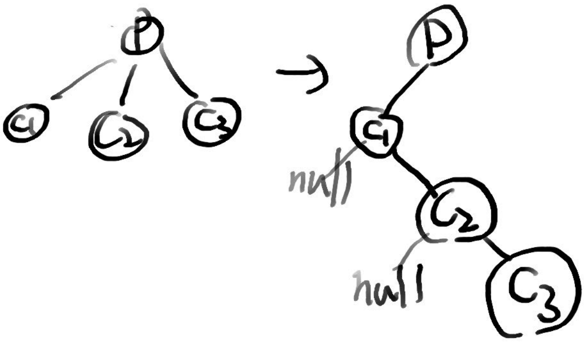
    B less clustering

Tree

gerenal → binary tree

↑ degree ≤ 2

left child, right sibling



binary tree → BST

↳ complete binary tree

↳ Heap O(1) → max / min

↳ priority queue

└ traversal : O(logn)

Graph

$G(V, Z)$

Adjacet.

(:)