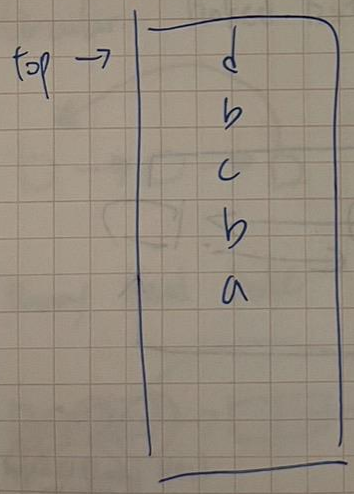
Quene
(First in First out)

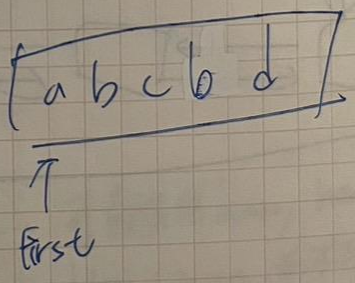
isEmpty() 是否為空

enquene() 新t會到後面

dequene() 刪除第一個

getFront() 取第一個





A word fitly spoken is like apples of gold in baskets of silver.
《Proverbs》

My Questions
Problems & Difficulties needing exploration

Array-based Implementation

back = (back+1) % MAX_QUEUE

會循環

假設 MAX_QUEUE = 3

back = 0 => (0+1) % 3 = 1

back = 1 => (1+1) % 3 = 2

back = 2 => (2+1) % 3 = 0

=> 0 -> 1 -> 2 -> 0 #

My Opinions
Thoughts, inspirations, and suggestions

Stack

(Last In first Out)

pop() 移除最上面

push() 新增到最上面

演算法效率

Time efficiency 時間效率

space efficiency 空間效率

big O 代表最大時間

$O(n^3) > O(n^2) > O(n\log_2 n) > O(n) > O(\log n) > O(1)$

效率高 ⟶ 效率低

並且是看最大的，忽略 constant

ex: $O(n^2+n) \Rightarrow O(n^2)$

ex: $O(n^3+(\log n) \Rightarrow O(n^3)$

ex: $O(\log n^2) \Rightarrow O(n^2)$

My Questions
Problems & Difficulties needing exploration

Slection sort :

① not stable

② worst : $O(n^2)$

③ best : $O(n^2)$

④ average : $O(n^2)$

1) 每次挑出被比較的資料中的最小或最大的放在頭或尾，因此big O每次都一樣。

My Opinions
Thoughts, inspirations, and suggestions

Bubble sort

① stable

② worst : $O(n^2)$

③ best : $O(n)$

④ average : $O(n^2)$

跟下一個資料比較，若大於 or 小於則交換，
每次會將最大或最小的資料放入頭 or 尾。

Merge sort

① stable

② worst O(nlogn)

③ best O(nlogn)

④ average O(nlogn)

用遞迴將資料做切半，最後再合併。

```
divide(a, begin, des){  int mid = (begin + des)/2
    if (begin < des)
        divide (a, begin, mid);
        divide (a, mid+1, des);
        merge (a, begin, mid, des);
}
```


Whenever it feels uncomfortable to tell the truth, that's often the most important time to tell it.
—Jennifer Lopez

My Questions
Problems & Difficulties needing exploration

Quick Sort

① not stable

② worst $O(n^2)$

③ best $O(n\log n)$

④ average $O(n\log n)$

用遞迴將資料分成基準點左邊右邊

My Opinions
Thoughts, inspirations, and suggestions

```
Quicksort ( a , low , high ) {
  if ( low < high ) {
    int pivot = partition ( butter, low, high )
    Quicksort ( a , low , pivot -1 );
    Quicksort ( a , pivot +1 , high );

  }// if
}
```

My Notes
Important Concepts worth keeping

Today: / /

Radix sort

① stable

② worst: LSD = O((n+k)xd)
穩定MSD = O(nxd)

③ best: O(n)

④ average: O(nxd)

先找出資料最大值，再以此值做為判斷位數的依據，直到除完後得到0。
依照除以十後的餘數進行一次排序，直到跑完最大位數的loop後，排序便完成。

My Questions
Problems & Difficulties needing exploration

Bindry tree

用 linked list 實作

```
struct Node {
    int a;
    node* right;
    node* left;
};

void Newnode (Node* &node, int a) {
    if (node == NULL)
        node = new Node;
        node -> a = a;
    }
    else if (node -> a > a)
        Newnode (node -> left, a);
    else if (node -> a < a)
        Newnode (node -> right, a);
```

My Opinions
Thoughts, inspirations, and suggestions

My Notes

Important Concepts worth keeping

Today: / /

求樹高：

用 recursion 算出左邊與右邊的高度
之後使比較誰較大，return 誰

操作二元樹時要特別注意指標是否為 NULL
指標操作不慎便會導致 error。

用不到的 pointer 必須要 delete。