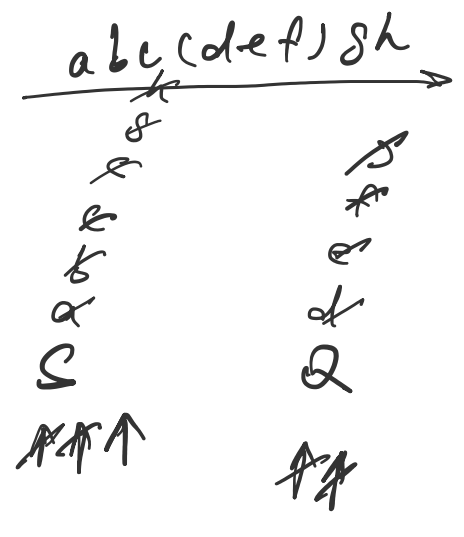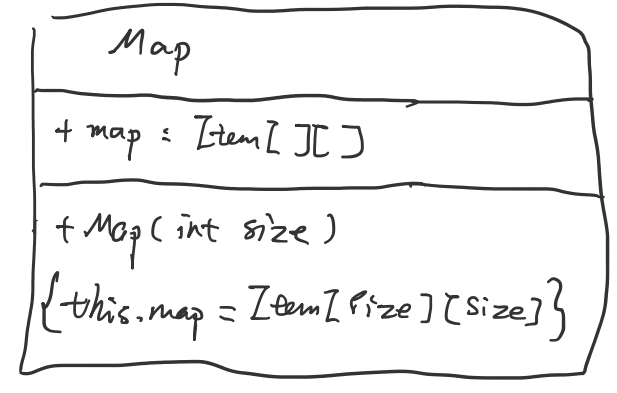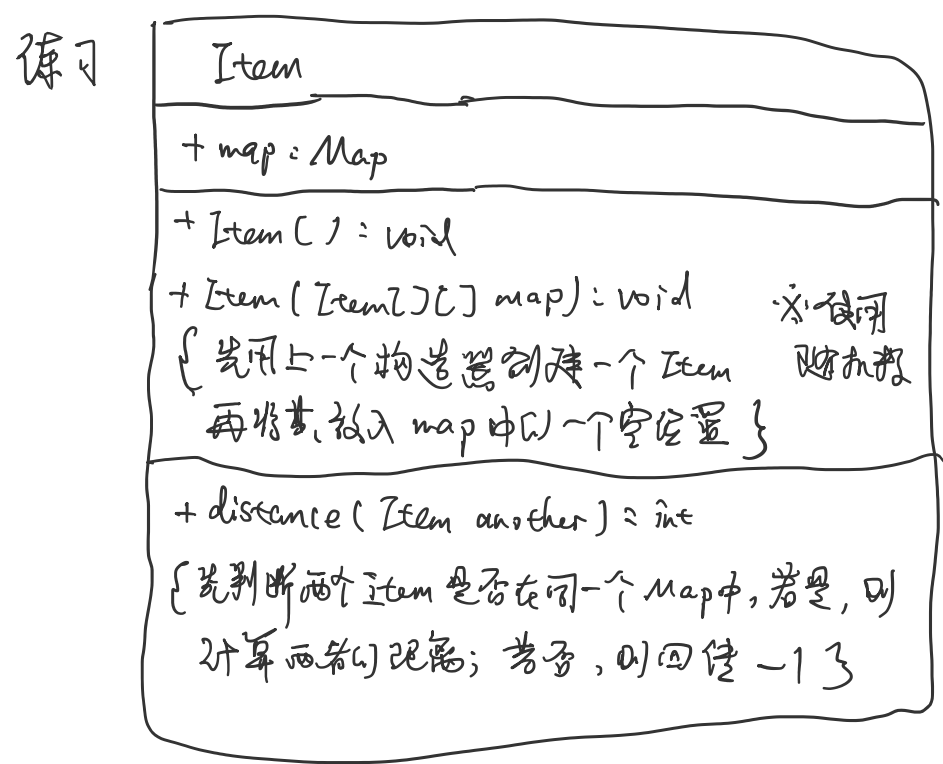作业： Input：一个包含 ( ) 的 String

output：倒序 input string，但 ( ) 作为一个整体 内部不被倒序

例： "abc (def) gh" → "hg (def) cba"

Stack : LIFO

Queue : FIFO

```
abc(def)gh
```

hgcdefjcba

练习

Item

+ map : Map

+ Item ( ) : void

+ Item (Item[][] map) : void
{ 先用上一个构造器创建一个 Item
再将书, 放入 map 中的一个空位置 }

+ distance (Item another) : int
{ 先判断两个 Item 是否在同一个 Map 中, 若是, 则
计算两个的距离; 若否, 则返回 -1 }

Map

+ map : Item[][]
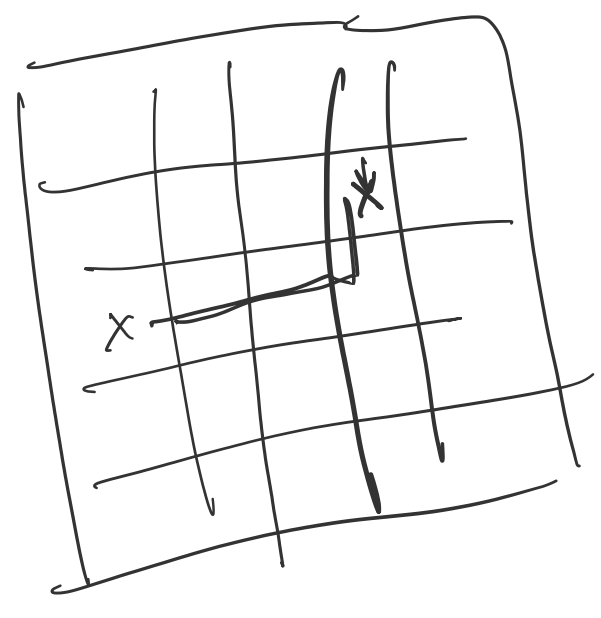
+ Map (int size)
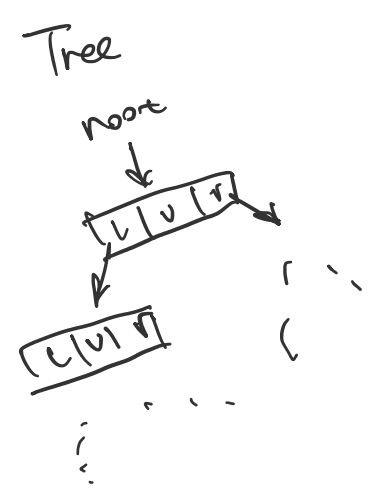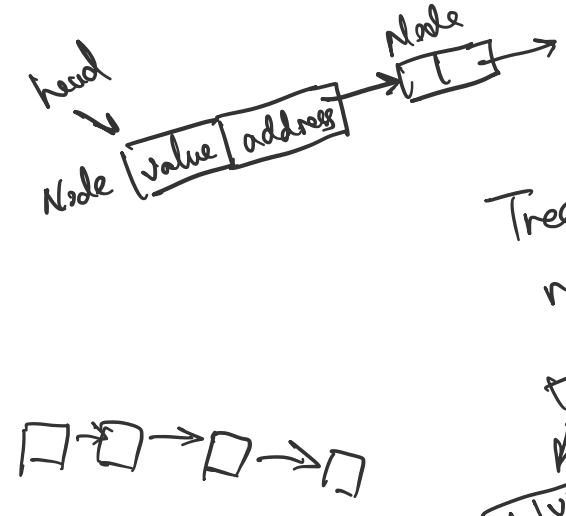{ this.map = Item[size][size] }

Main 中完成 :

Map map = new Map (10)

Item i1 = new Item (map)

Item i2 = new Item (map)

Item i3 = new Item (null)

Sout (i1. distance (i2))

Sout (i1. distance (i3))

内容： 1. ArrayList : add(), get(int index), set(int index, Object obj)

remove(int index), size(), indexOf(Object obj)

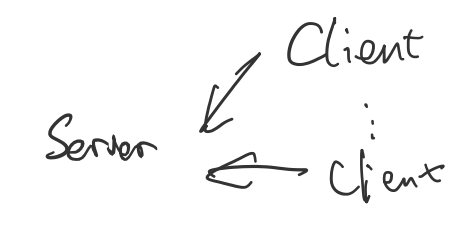contains(Object obj), clone(), subList(int from, int to)

sort()　　　　左闭右开
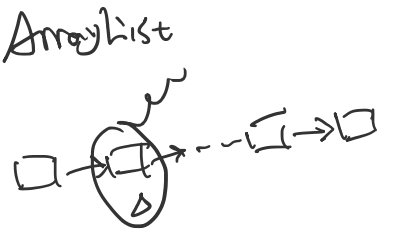
2. HashSet : 同 ArrayList, 但不允许重复元素

3. HashMap : 是一个 key - Value Set : put(key, value), get(key), remove(key)

containsKey(key), containsValue(), putIfabsent(…, …), clone(), keySet()

forEach()

交化型

学 lambda

```
1 2 3 4 5 4 3 2 1 2 3 4 5
```

IndexOf (5) → 4

LastOf (5)

key) → Value

Client
Server
Client

ArrayList

Map<T>

public T[][] field

C　　100 86　　S

70 87

20 174