7, 1, 3, 7, 3, 7

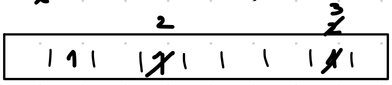
① Sorting algorithms

w.c. $\Theta(n \log n)$

$\boxed{\le}$

- merge sort → extra memory n
- heap sort → not a stable algorithm

② $x_i \in \mathbb{Z}$ $x_i \in [0, N-1]$



1, 3, 3, 7, 7, 7

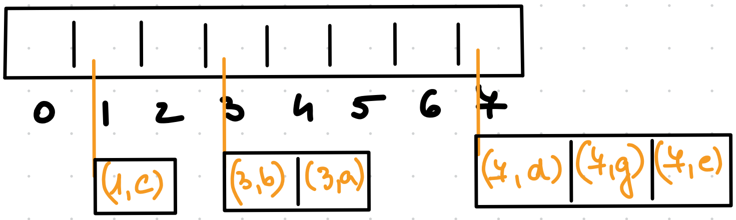
$\Theta(n+N)$

* when sorting based on a relation, the best complexity we can get is nlogn

Bucket Sort

given : a sequence S
with elems of type pair (key, value)
and keys are integers from [0, N)
sort S (using ...)

ex: (7, d) (1, c) (3, b) (7, g) (3, a) (7, e) N=8



```
DT   sequence
        initEmpty (S)
        isEmpty (S) ⇒ true / false
        first (S) ⇒ e
        removeFirst (S)
        addLast (S,e)
```

```
subalg   BucketSort (S, N)
   @ allocate B- array with N positions
       each element being a sequence
   for i ← 0,N-1 execute
       initEmpty (B[i])
   end for
   while ¬isEmpty (S)
       <k,v> ← first (S)
       addLast (B, <k,v>)
       removeFirst (S)
   end while
```

```
for i ← 0, N-1 execute
    while ¬ is Empty (B[i])
        <k,v> ← first(B[i])
        add Last (S, <k,v>)
        remove First (B[i])
    end while
end for
End subalg
```

$\longrightarrow \Theta(n+N)$

Lexicographic sort

(a, c, r)   (c, a, r)   (r, a, c)

* tuples with dim 3

(1, 2, 3)   (1, 1, 3)   (2, 1, 1)

⇒ sorted (1, 1, 3)   (1, 2, 3)   (2, 1, 1)

"≤" relation

tuple with dim d

$(x_1, x_2, \dots, x_n)$ "≤" $(y_1, y_2, \dots, y_n)$ ⟺ $(x_1 < y_1)$ OR $(x_1 = y_1$ AND $(x_2, \dots, x_d) \le (y_2, \dots, y_d))$

→ R relation over tuples of dimention d

$R = (R_1, \dots, R_d)$ R_i -is the relation between x_i and y_i

```
subalgorithm Lexicographic Sort (S, d, R)
    for i ← d, 1, -1 execute
        stable Sort (S, R_i)
    end For
end subalg.
```

(1, 2, $\acute{3}$)   (1, 1, $\acute{3}$)   (2, 1, $\acute{1}$)

sort by i=3

(2, 1, 1)   (1, $\acute{2}$, 3)   (1, $\acute{1}$, 3)

i=2

(2, 1, 1)   (1, 1, 3)   (1, 2, 3)

i=1

(1, 1, 3)   (1, 2, 3)   (2, 1, 1)

by i=1

($\overset{\downarrow}{1}$, 2, 3)   ($\overset{\downarrow}{1}$, 1, 3)   ($\overset{\downarrow}{2}$, 1, 1)

by i=2

(1, $\overset{\downarrow}{1}$, 3)   (2, $\overset{\downarrow}{1}$, 1)   (1, $\overset{\downarrow}{2}$, 3)

by i=3

(2, 1, 1)   (1, 1, 3)   (1, 2, 3)

Redix Sort algorithm

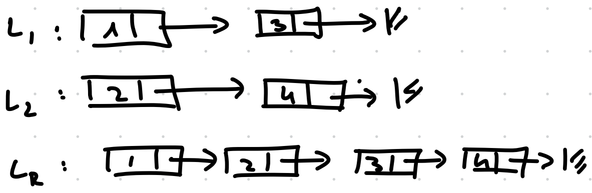
- lexicographic sort using a stable sort algorithm

Implement merge algorithm over 2 SLL

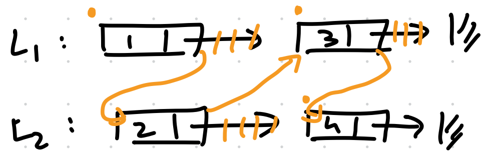
```
SLNode
    info  : TComp
    next  : ↑SLNode

SLL
    head  : ↑SLNode
```

V_1) Create a new list with all the elements



$\Theta(m+n)$



```
LR :    head LR
        tail LR
        c1
        c2
```

```
Subalg    merge (L1, L2, LR)
    head LR ← NIL
    tail LR ← NIL
    c1 ← L1.head
    c2 ← L2.head
```

BC : $\Theta(\min(m,n))$

WC : $\Theta(m+n)$