240500 1032 - Danny IDUKUNDATWEJE

Data structure and Algorithm

16/02/2025 Sunday

ASSIGNMENT

Q. Numbers = [29, 14, 37, 13, 105, 42, 7, 95, 22]

(i). (a). **Bubble sort**

n = 9

For k = 1 to 8, (n - 1)

1st pass: j = 0 to (n - k - 1) = 7

j=0; 29 > 14, swap ⇒ [14, 29, 37, 13, 105, 42, 7, 95, 22]

j=1; 29 < 37, no swap

j=2; 37 > 13, swap ⇒

0	1	2	3	4	5	6	7	8
14	29	13	37	105	42	7	95	22

j=3; 37 < 105, no swap

j=4; 105 > 42, swap ⇒

0	1	2	3	4	5	6	7	8
14	29	13	37	42	105	7	95	22

j=5; 105 > 7, swap ⇒

0	1	2	3	4	5	6	7	8
14	29	13	37	42	7	105	95	22

j=6; 105 > 95, swap ⇒

0	1	2	3	4	5	6	7	8
14	29	13	37	42	7	95	105	22

j=7; 105 > 22, swap ⇒

0	1	2	3	4	5	6	7	8
14	29	13	37	42	7	95	22	105

2nd pass (k = 2): j = 0 to (n - k - 1) = 6

j=0; 29 > 14, no swap

j=1; 29 > 13, swap ⇒

0	1	2	3	4	5	6	7	8
14	13	29	37	42	7	95	22	105

j=2; 37 > 29, no swap

j=3; 42 > 37, no swap

j=4; 42 > 7, swap ⇒

0	1	2	3	4	5	6	7	8
14	13	29	37	7	42	95	22	105

j=5, 95 > 42, no swap.

j=6, 95 > 22, swap =>

0	1	2	3	4	5	6	7	8
14	13	29	37	7	42	22	95	105

3rd pass (k = 3): j = 0 to (n - k - 1) = 5

j=0; 14 > 13, swap =>

0	1	2	3	4	5	6	7	8
13	14	29	37	7	42	22	95	105

j=1, 29 > 14, no swap

j=2, 37 > 29, no swap

j=3, 37 > 7, swap =>

0	1	2	3	4	5	6	7	8
13	14	29	7	37	42	22	95	105

j=4, 42 > 37, no swap

j=5, 42 > 22, swap =>

0	1	2	3	4	5	6	7	8
13	14	29	7	37	22	42	95	105

4th pass (k = 4): j = 0 to (n - k - 1) 4

j=0, 14 > 13, no swap

j=1, 29 > 14, no swap

j=2, 29 > 7, swap =>

0	1	2	3	4	5	6	7	8
13	14	7	29	37	22	42	95	105

j=3, 37 > 29, no swap

j=4, 37 > 22, swap =>

0	1	2	3	4	5	6	7	8
13	14	7	29	22	37	42	95	105

5th pass (k = 5): j = 0 to (n - k - 1) 3

j=0, 14 > 13, no swap

j=1, 14 > 7, swap =>

0	1	2	3	4	5	6	7	8
13	7	14	29	22	37	42	95	105

j=2, 29 > 14, no swap

j=3, 29 > 22, swap =>

0	1	2	3	4	5	6	7	8
13	7	14	22	29	37	42	95	105

~~j=4, 37 > 29, no swap~~

6th pass (k = 6): j = 0 to (n - k - 1) 2

j=0, 13 > 7, swap =>

0	1	2	3	4	5	6	7	8
7	13	14	22	29	37	42	95	105

j=1 = 14 > 13, no swap

j=2, 29 > 22, no swapp

Henceforth:

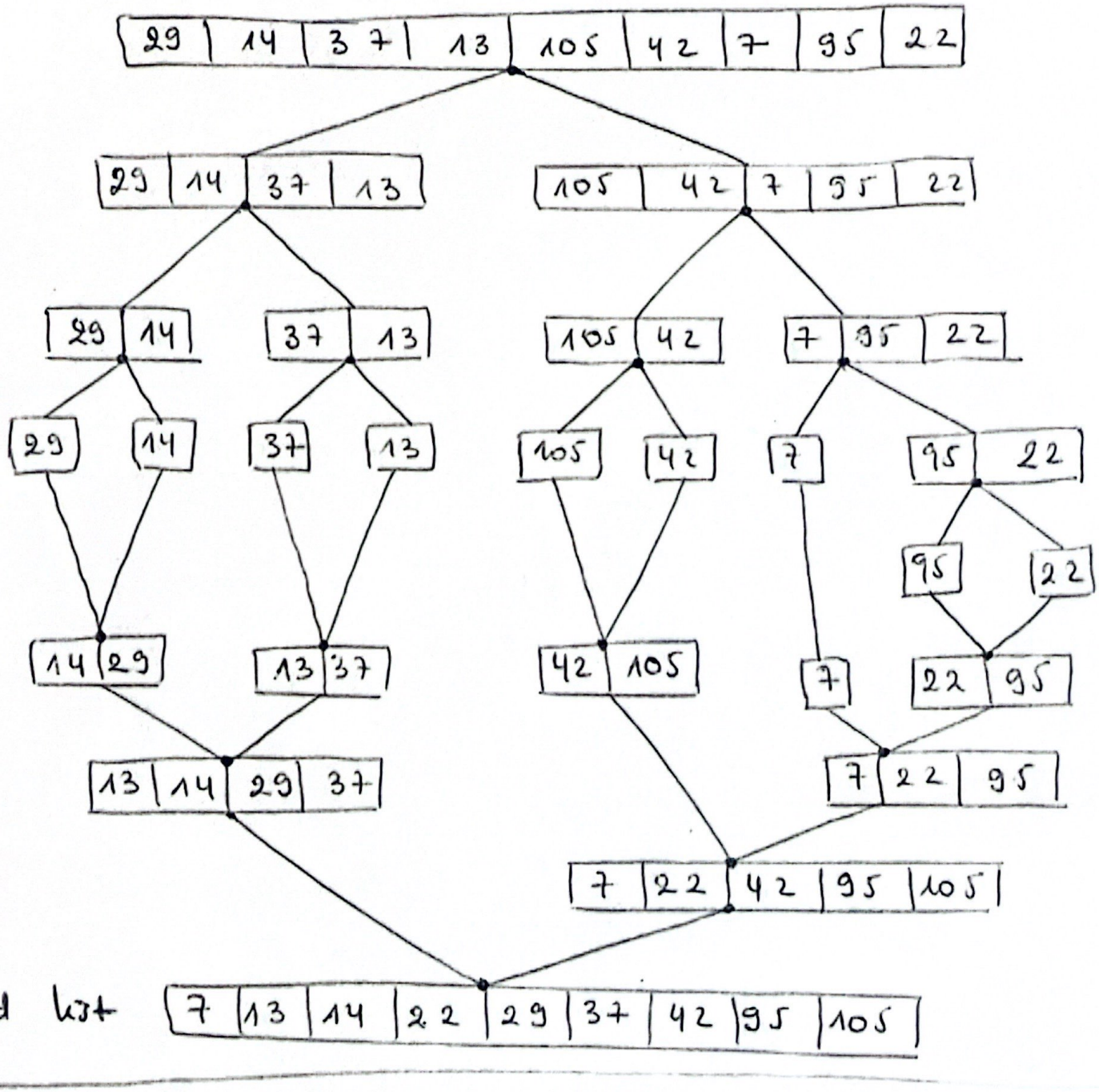
sorted list:

↳ numbers = [7, 13, 14, 22, 29, 37, 42, 95, 105]

---

b) Merge sort

n = 9 mid = $\frac{9}{2}$ = 4.5 (mid ≃ 4)



---

c) Selection sort

n = 9

For i = 0 to (n-1) = 8

$j = (i+1) = 1$ to $(n-1) = 8$

⑧. A[0] = 29

29	14	37	13	105	42	7	95	22

A[smallsub] = 7

A[0] > A[smallsub]

A[0] > 7 (True) – swap.

7	14	37	13	105	42	29	95	22

⑨. A[1] = 14

7	14	37	13	105	42	29	95	22

A[smallsub] = 13

14 > 13 (True) – swap

7	13	37	14	105	42	29	95	22

⑩. A[2] = 37

7	13	37	14	105	42	29	95	22

A[smallsub] = 14

37 > 14 (True) - swap

7	13	14	37	105	42	29	95	22

⑪. A[3] = 37

7	13	14	37	105	42	29	95	22

A[smallsub] = 22

37 > 22 (True) - swap

⑫. ~~A[4] < 10~~

7	13	14	22	105	42	29	95	37

⑬. A[4] = 105

7	13	14	22	105	42	29	95	37

A[smallsub] = 29

105 > 29 (True) - swap

7	13	14	22	29

42	105	95	37

⑤. A[5] = 42

7	13	14	22	29	42

105	95	37

A[smallsub] = 37

42 > 37 (True) - swap

7	13	14	22	29	37

105	95	42

⑥. A[6] = 105

7	13	14	22	29	37	105

95	42

A[smallsub] = 42

105 > 42 (True) - swap

7	13	14	22	29	37	42

95	105

⑦. A[7] = 95

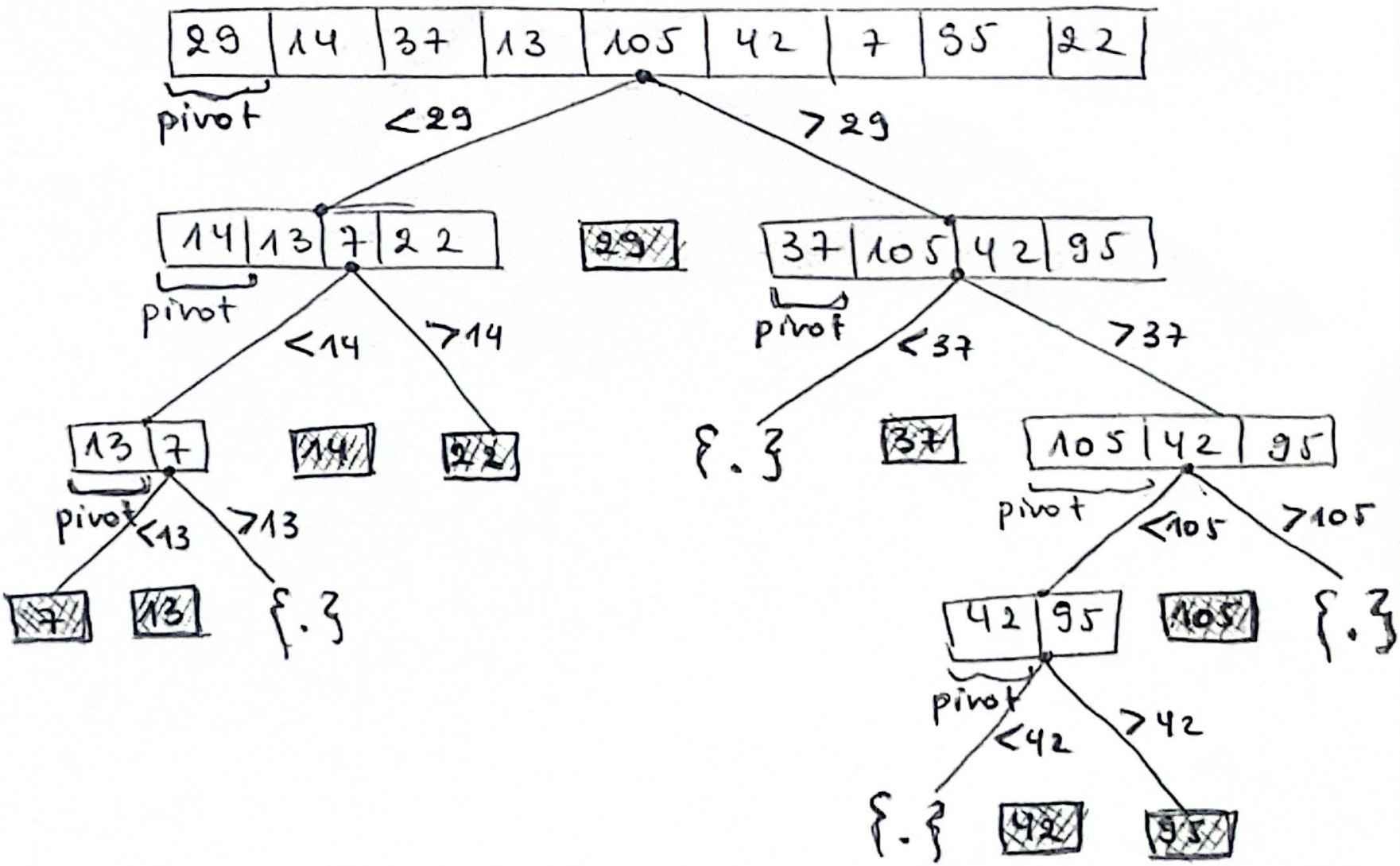
A[7] > A[smallsub]

95 < 105 (True), no swap

Henceforth, the sorted list:

or.

d) Quick sort (n = 9)



Henceforth, sorted array:

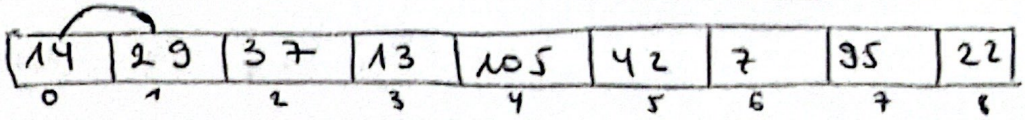
~~29~~ 7 13 14 22 29 37 42 95 105

2) **Insertion sort** $(n = 9)$

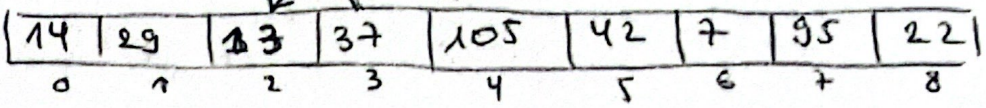
for $i = 1$ to $(n-1) = 8$: $j = i$

while $(j > 0)$ and $A[j] < A[j-1]$

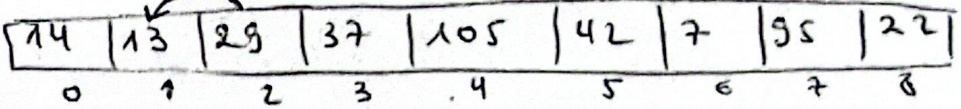
- $j = 1$, $29 > 14$, swap: temp = $A[1] = 14$
  $A[1] = 29$
  $A[0] = 14$



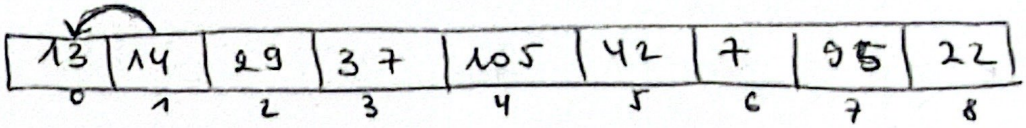
- $j = 2$, $37 > 29$, no swap
- $j = 3$, $37 > 13$, swap:



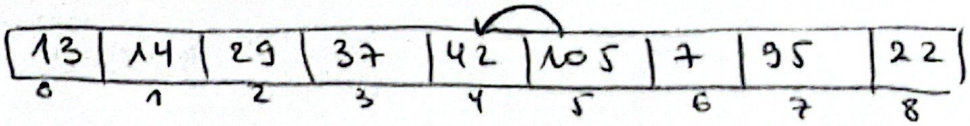
$29 > 13$, swap:



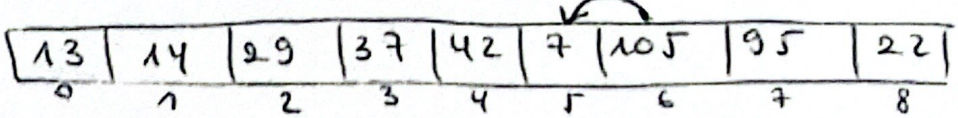
$14 > 13$, swap:



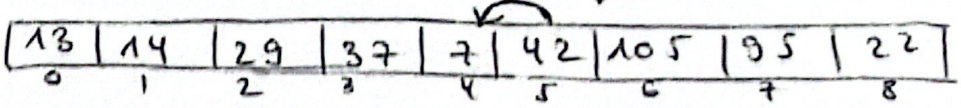
- $j = 4$, $105 > 37$, no swap
- $j = 5$, $105 > 42$, swap:



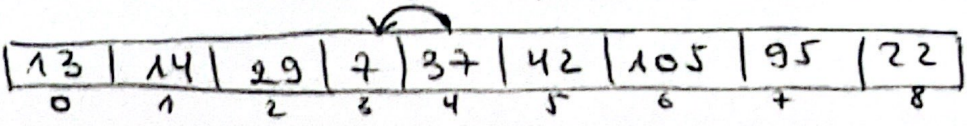
- $j = 6$, $105 > 7$, swap:



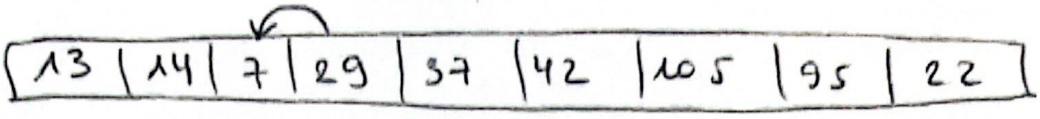
$42 > 7$, swap:



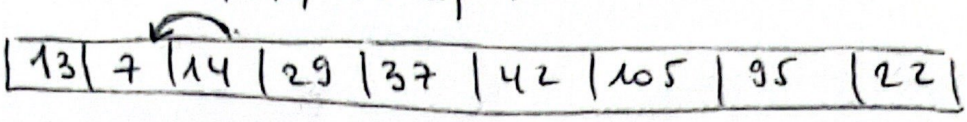
$37 > 7$, swap:



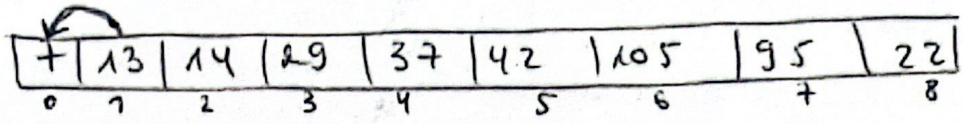
29 > 7, swap :



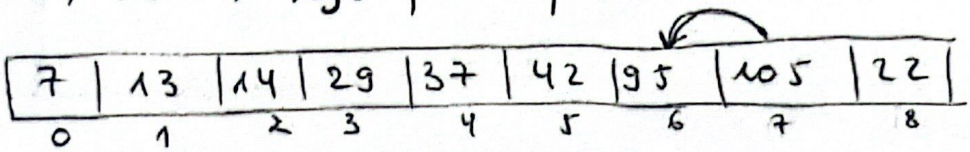
14 > 7, swap :



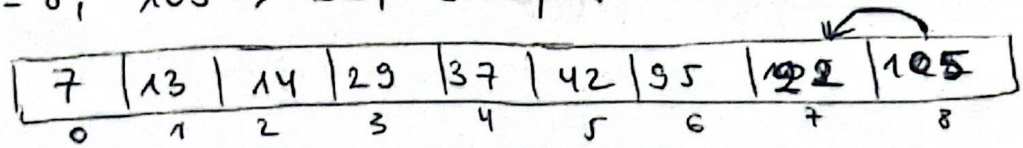
13 > 7, swap :



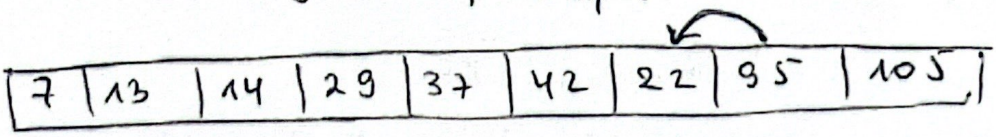
⑦ j = 7, 105 > 95, swap :



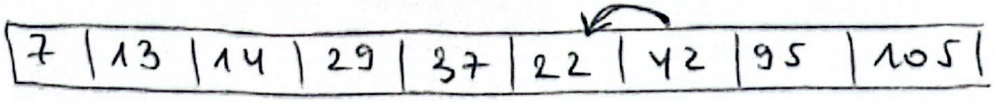
⑧ j = 8, 105 > 22, swap :



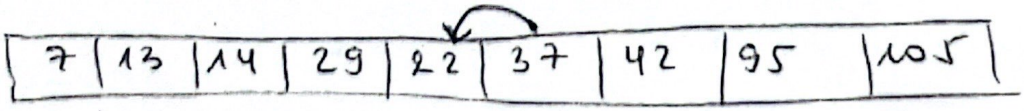
95 > 22, swap :



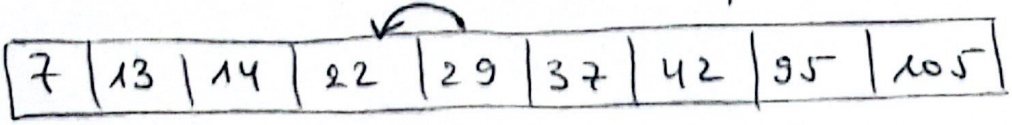
42 > 22, swap :



37 > 22, swap :

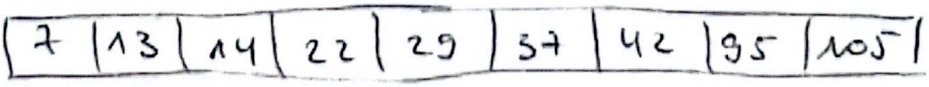


29 > 22, swap :



14 < 22, no swap

Hence sorted list :



End

(ii). Complexity comparison

type of algorithm	time complexity	space complexity
a) Bubble sort	$O(9^2) = O(81)$	$O(1)$
b) Merge sort	$O(9 \log 9) \simeq O(8.6)$	$O(9)$
c) selection sort	$O(9^2) = O(81)$	$O(1)$
d) Quick sort	$O(9 \log 9) \simeq O(8.6)$	$O(\log 9) \simeq O(0.9)$
e) Insertion sort	$O(9^2) = O(81)$	$O(1)$

Where n = number of item in array.

Comment:

Overall, Merge and Quick sort has minimum time complexity where as bubble, selection and Insertion has minimum space complexity.

As result, Quick sort has best time complexity of $O(9 \log 9) \simeq O(8.6)$ and space complexity of $O(\log 9) \simeq O(0.9)$