

[6, 7, 2, 8, 5, 1, 3, 4]

$$\begin{array}{|c|c|c|c|} \hline 6 & 7 & 2 & 8 \\ \hline \end{array} \quad \begin{array}{|c|c|c|c|} \hline 5 & 1 & 3 & 4 \\ \hline \end{array}$$

9
7
7

100

$$\begin{bmatrix} 2 & 1 & 3 \\ 3 & 1 & 3 \end{bmatrix}$$

4

(2, 17)

[2] [1]

$$\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 1 \\ 1 \\ 2 \\ 3 \end{bmatrix}$$

4

him

卷四

 $[6, 7, 25, 13, 4]$
$$[6, 7, 2, 5] \quad [1, 3]$$

[6, 7]

67 [7]

6, 7

256

1234567



由 扫描全能王 扫描创建

Qian Zhou
APCS2 pd 01

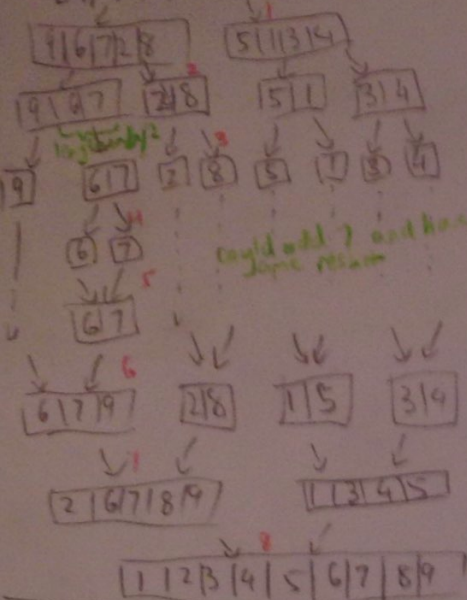
HW-06 How fast are your turtles?

2018-02-13

Previous Paper: II:4; VII:6; VIII:6

Régine IX:8

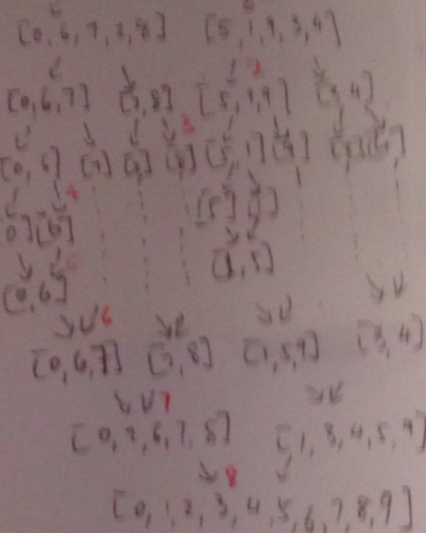
[9, 6, 7, 2, 8, 5, 1, 3, 4]



could add 7 and have some result

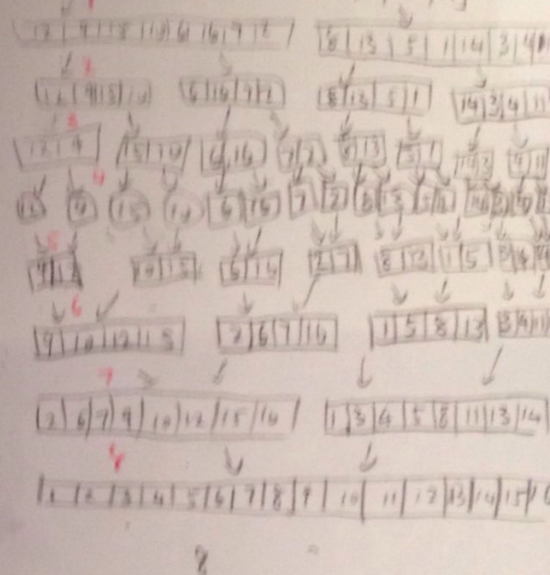
Richard X:8

[0, 6, 7, 2, 8, 5, 1, 9, 3, 4]



William XII:8

[12, 9, 13, 10, 4, 14, 7, 2, 8, 13, 5, 1, 14, 3, 4, 11]



$n: 7$

I: 1

II: 2

III-IV: 4

V-VII: 6

IX-XVI: 8

XVII-XXXII: 10

if $n = I; \frac{n}{2} = 1$
if $n = II; \frac{n}{2} = 2$
else

$(\lceil \log_2 n \rceil \cdot 2) = 7$

because according to this diagram, the time of certain length arrays increases by multiples of 2, and the upper/lower bounds of such values remain close to exponents of 2.

Perhaps $\log_2 n$ determines the maximum of numbers the green numbers in Régine's code indicate, such that the total layers remain constant.

Each layer is $O(n)$ run time, at least one that is full, since each needs to either make two new arrays which would require copying elements into each, or merging them which would require amortizing and copying elements into the sorted array, both of which require linear runtime.

For n length, $\lceil \log_2 n \rceil \cdot 2$ layers are needed, so the total runtime is about $O(n \log_2 n)$.



HW-06 New just are given

2018-02-13

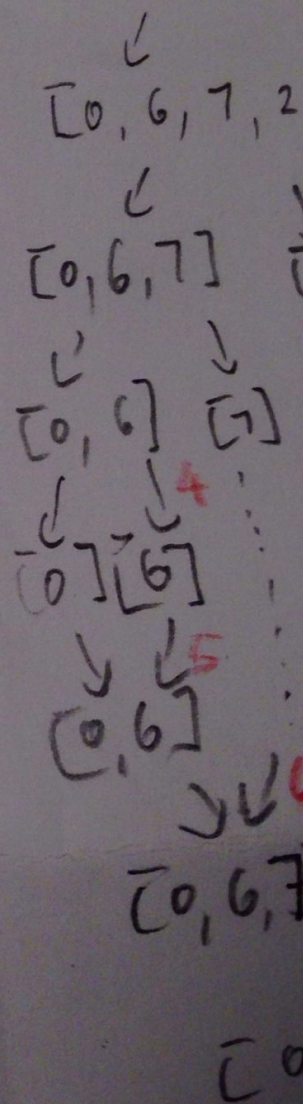
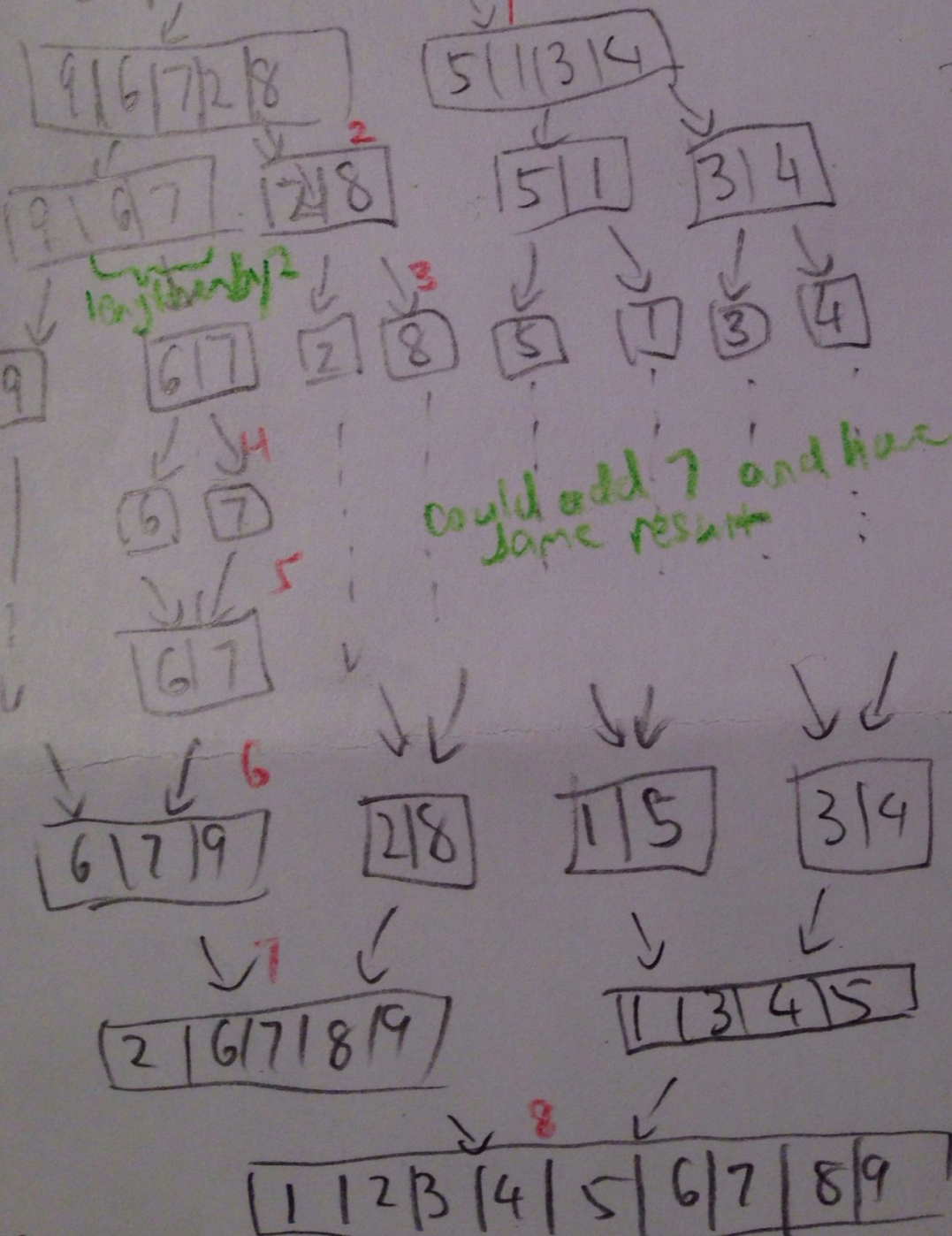
Previous Paper: $\text{II} : 4$; $\text{VII} : 6$

Régime IX : 8

Richard

[9, 6, 7, 2, 8, 5, 1, 3, 4]

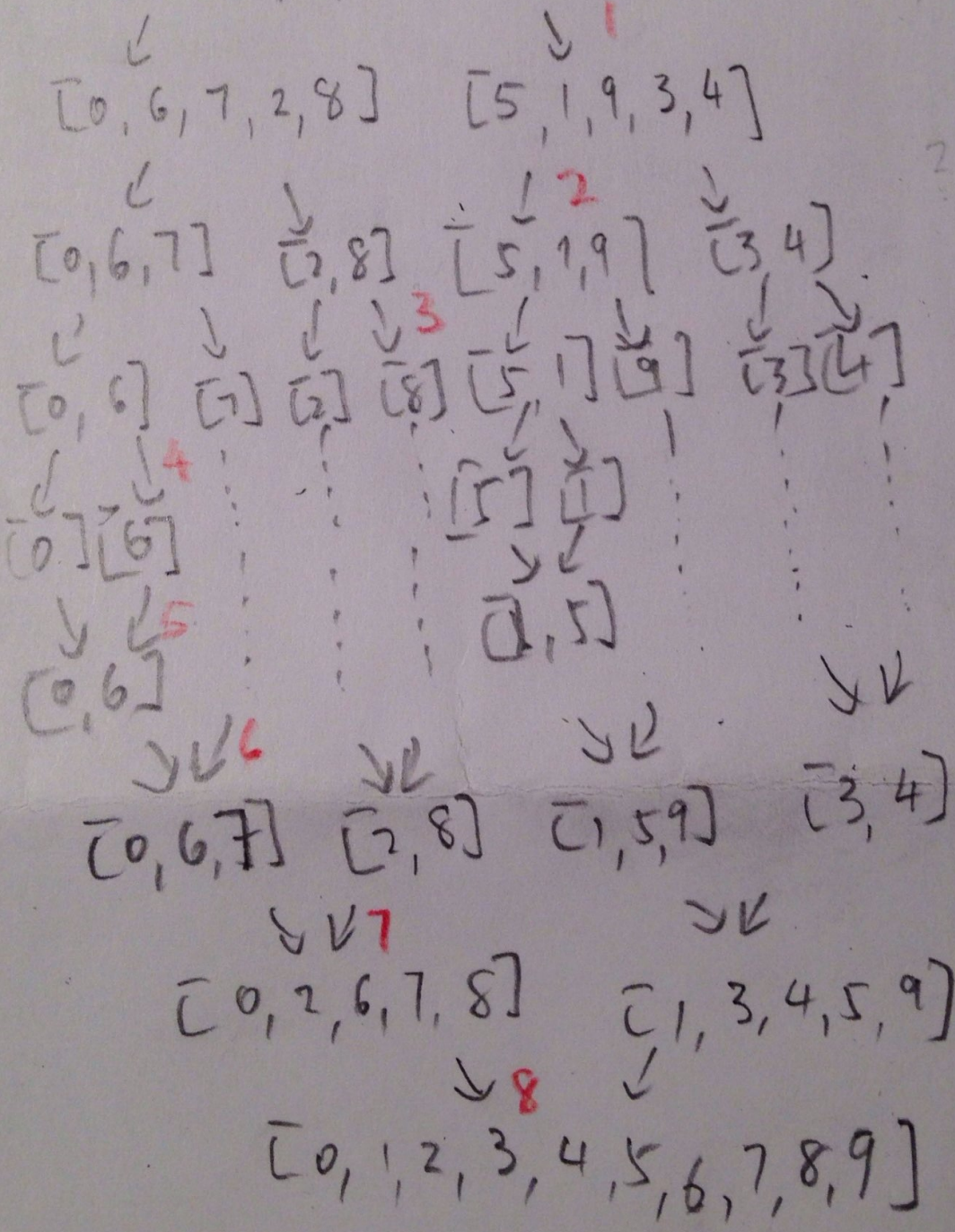
[0, 6, 7, 2]



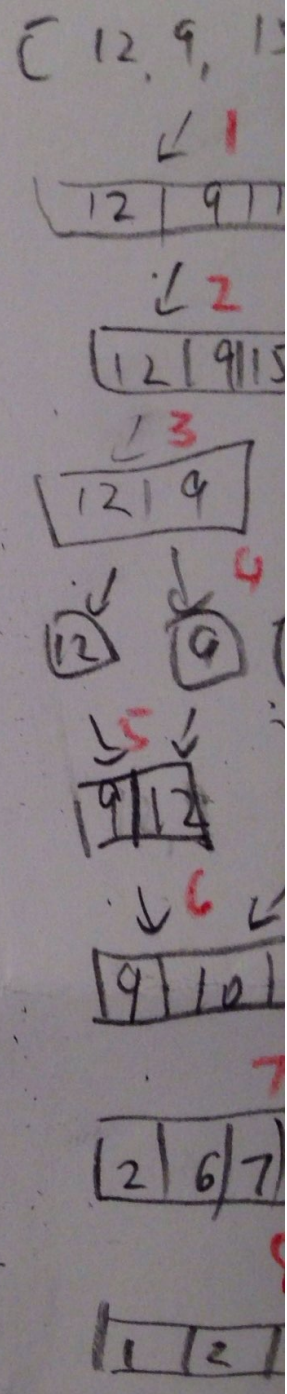
IV:4 ; VII:6 ; VIII:6

Richard X:8

[0,6,7,2,8,5,1,9,3,4]



William
~~XVI~~:8



William

~~XIV~~ : 8

[12, 9, 15, 10, 6, 16, 7, 2, 8, 13, 5, 1, 14, 3, 4, 11]

1

12 | 9 | 15 | 10 | 6 | 16 | 7 | 2

8 | 13 | 5 | 1 | 14 | 3 | 4 | 11

2

12 | 9 | 15 | 10

6 | 16 | 7 | 2

8 | 13 | 5 | 1

14 | 3 | 4 | 11

3

12 | 9

15 | 10

6 | 16

7 | 2

8 | 13

5 | 1

14 | 3

4 | 11

4

12

9

15

10

6

16

7

2

8

13

5

1

5

9 | 12

10 | 15

6 | 16

2 | 7

8 | 13

1 | 5

3 | 14

6

9 | 10 | 12 | 15

2 | 6 | 7 | 16

1 | 5 | 8 | 13

3 | 4 | 11

7

2 | 6 | 7 | 9 | 10 | 12 | 15 | 16

1 | 3 | 4 | 5 | 8 | 11 | 13 | 14

8

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16

8



$n : 7$
 $I : 1$
 $II : 2$
 $III-IV : 4$
 $V-VIII : 6$
 $IX-XVI : 8$
 $XVII-XXII : 10$

if $n = I : t = 1$
 if $n = II : t = 2$
 else

$$\lceil \log_2 n \rceil \cdot 2 = t$$

because according to this diagram, the
 size of certain length arrays increases
 by multiples of 2, and the upper lower
 bounds of such values remain close to
 exponents of 2.

Perhaps $\log_2 n$ determines the maximum of numbers
 the green comments on Régnie's code indicate,
 such that the total layers remain constant.

Each layer is $O(n)$ run time, at least one that is full,
 since each needs to either make two new arrays which would require
 copying elements into each, or merging them which would require
 complexity and copying elements into the sorted array, both of which
 require linear runtime.

For n length, $\lceil \log_2 n \rceil \cdot 2$ layers are needed, so the total
 runtime is about $O(n \log_2 n)$

