THE UNIVERSITY OF
MELBOURNE

COMP90038
Algorithms and Complexity

Assignment Project Exam Help

Lecture 11: S

https://eduassistpro.github.io/

and-Conquer

(with thanks to Hara ergaard)

Add WeChat edu_assist_pro

Toby Murray

✉ toby.murray@unimelb.edu.au

DMD 8.17 (Level 8, Doug McDonell Bldg)

http://people.eng.unimelb.edu.au/tobym

@tobycmurray

Divide and Conquer

THE UNIVERSITY OF MELBOURNE

- We earlier studied recursion as a powerful problem solving technique.

- The **divide-and-conquer** strategy tries to make the most of this idea:

Assignment Project Exam Help

1. Divide the given into smaller instances.

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

2. Solve the smaller instances ely.

3. Combine the smaller solutions to solve the original instance.

- This works best when the smaller instances can be made to be of equal (or near-equal) size.

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Split-Solve-and-Join Approach

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Divide and Conquer Algorithms

- We will discuss:
  - The Master Theorem
  - Mergesort
  - Quicksort
  - Tree traversal
  - Closest Pair revisited

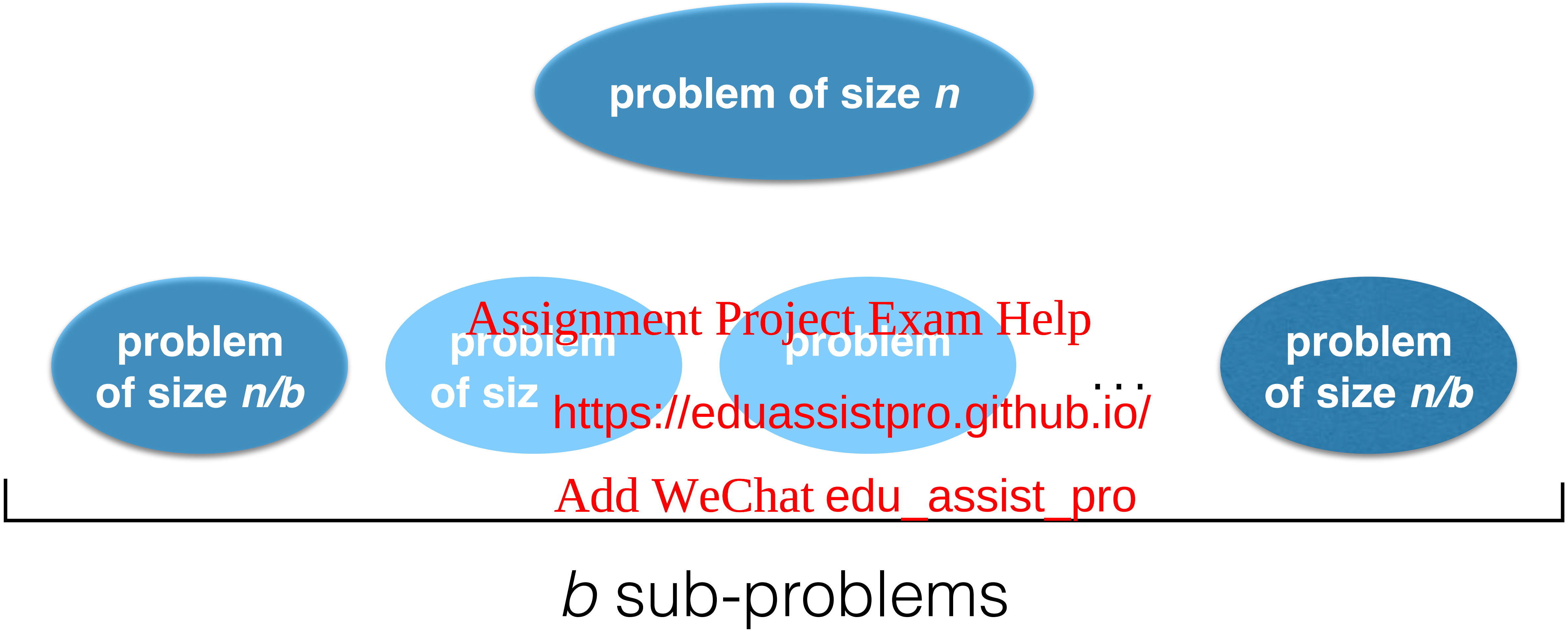
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

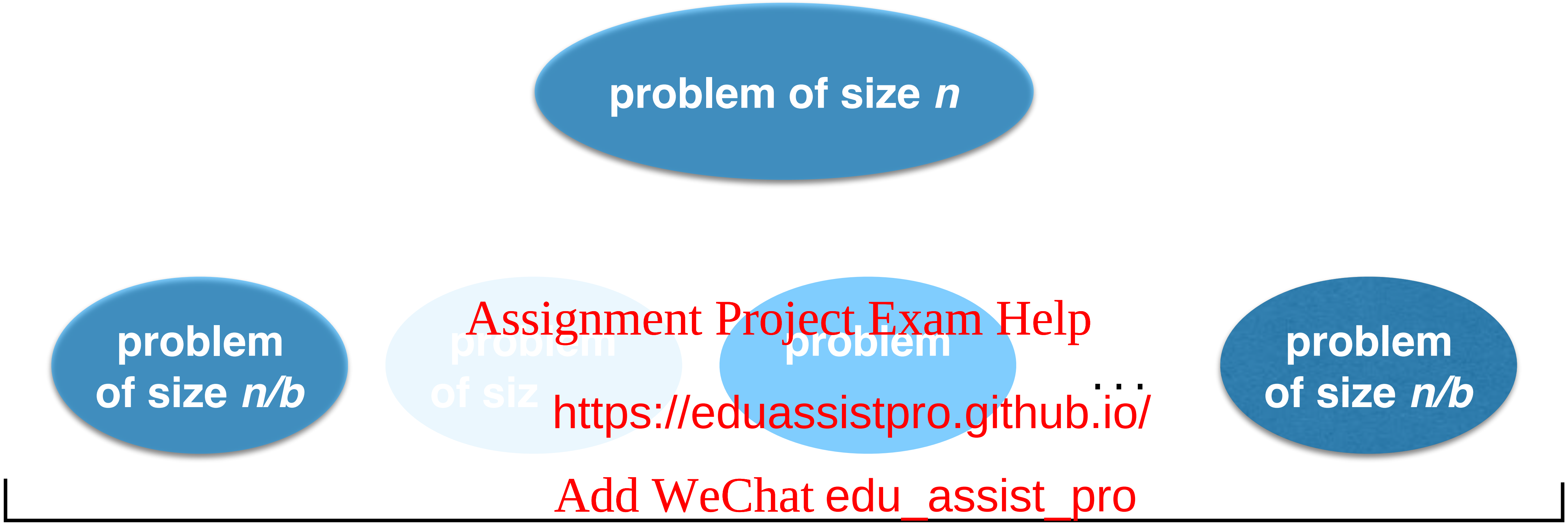
Divide-and-Conqer General Case



Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Divide-and-Conqer General Case



problem of size *n*

problem of size *n/b*

problem of siz

problem

. . .

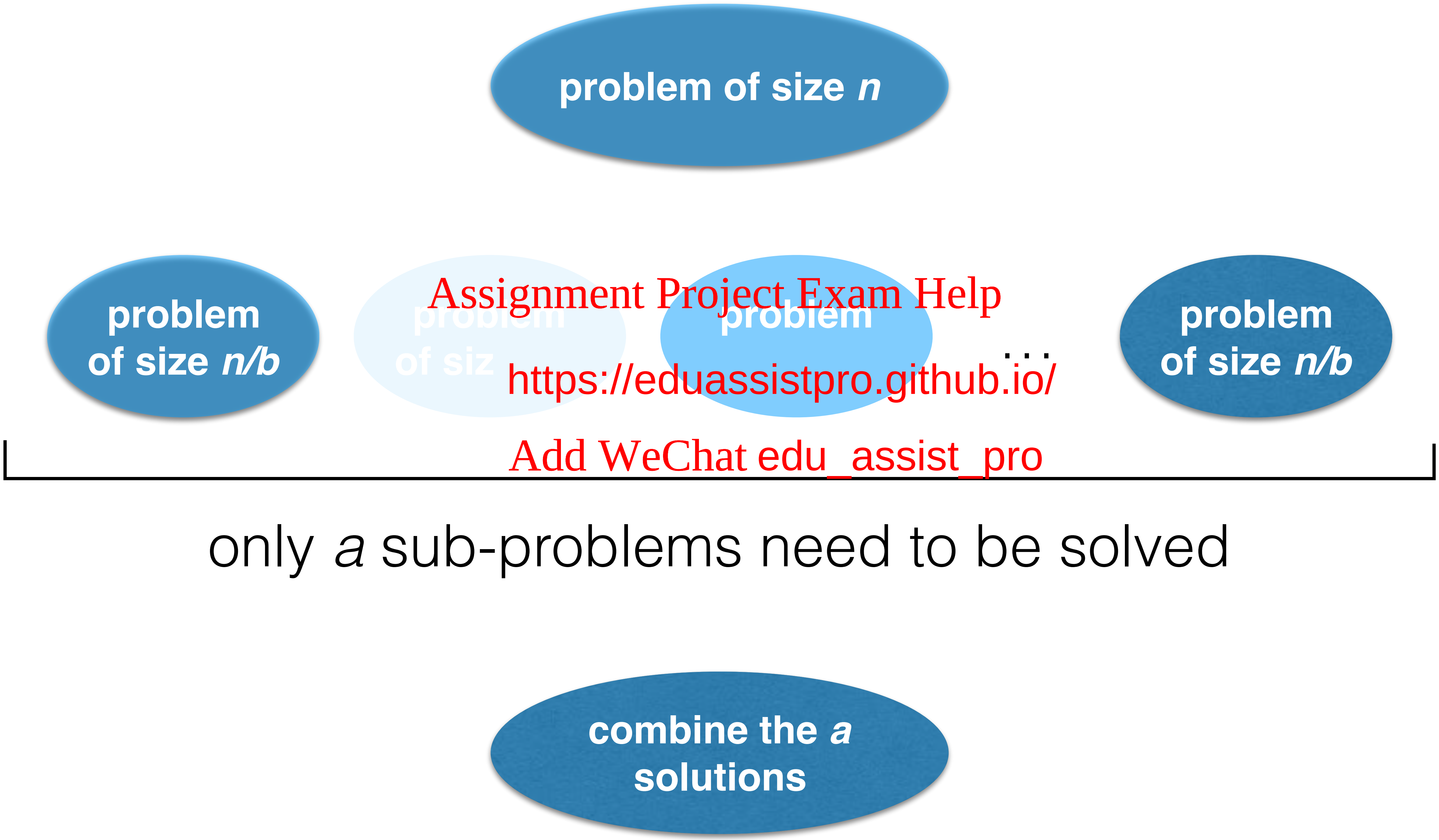
problem of size *n/b*

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

only *a* sub-problems need to be solved

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Divide-and-Conqer General Case



problem of size *n*

problem of size *n/b*

problem of siz

problem

. . .

problem of size *n/b*

only *a* sub-problems need to be solved

combine the *a* solutions

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Divide-and-Conquer Recurrences

- What is the time required to solve a problem of size *n* by divide-and-conquer?

- For the general case, assume we split the problem into *b* instances (each of size *n/b*), of which *a* need to be solved:

$$T(\qquad n)$$

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

where *f*(*n*) expresses the time spent on dividing a problem into *b* sub-problems and combining the *a* results.

- (A very common case is $T(n) = 2T(n/2) + n$.)

- How to find closed forms for these recurrences?

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

The Master Theorem

- (A proof is in Levitin's Appendix B.)

- For integer constants $a \geq 1$ and $b > 1$, and function f with $f(n) \in \Theta(n^d)$, $d \geq 0$, the recurrence

Assignment Project Exam Help

$$T(\qquad n)$$

https://eduassistpro.github.io/

(with T(1) = c) has solution

Add WeChat edu_assist_pro

$$T(n) = \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log n) & \text{if } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

- Note that we also allow *a* to be greater than *b*.

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 1

$T(n) = 2T(n/2) + n$ a = 2, b = 2, d = 1

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 1

$T(n) = 2T(n/2) + n$

$a = 2, b = 2, d = 1$

$a = b^d$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

So, by the Master Theorem, $T(n) \in \Theta(n \log n)$

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 1

$T(n) = 2T(n/2) + n$ $a = 2, b = 2, d = 1$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 1

$T(n) = 2T(n/2) + n$ a = 2, b = 2, d = 1

$T(n) = 2(2T(n/4) + (n/2)) + n$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 1

$T(n) = 2T(n/2) + n$ a = 2, b = 2, d = 1

$T(n) = 4T(n/4) + 2(n/2) + n$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 1

$T(n) = 2T(n/2) + n$ a = 2, b = 2, d = 1

$T(n) = 4(2T(n/8) + n/4) + 2(n/2) + n$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 1

$T(n) = 2T(n/2) + n$ a = 2, b = 2, d = 1

$T(n) = 8T(n/8) + 4(n/4) + 2(n/2) + n$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 1

$T(n) = 2T(n/2) + n$ a = 2, b = 2, d = 1

$T(n) = 8(2T(n/16) + n/8) + 4(n/4) + 2(n/2) + n$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 1

$T(n) = 2T(n/2) + n$ a = 2, b = 2, d = 1

$T(n) = 16T(n/16) + 8(n/8) + 4(n/4) + 2(n/2) + n$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 1

$T(n) = 2T(n/2) + n$ a = 2, b = 2, d = 1

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 1

$T(n) = 2T(n/2) + n$ 　　　　 a = 2, b = 2, d = 1

$T(n) \in \Theta(n \log n)$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 2

$T(n) = 4T(n/4) + n$

a = 4, b = 4, d = 1

$a = b^d$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

So, by the Master Theorem, $T(n) \in \Theta(n \log n)$

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 2

$T(n) = 4T(n/4) + n$ $\qquad$ $a = 4, b = 4, d = 1$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 2

$T(n) = 4T(n/4) + n$ a = 4, b = 4, d = 1

$T(n) = 4(4T(n/16) + (n/4)) + n$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 2

$T(n) = 4T(n/4) + n$ a = 4, b = 4, d = 1

$T(n) = 16T(n/16) + 4(n/4) + n$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 2

$T(n) = 4T(n/4) + n$ a = 4, b = 4, d = 1

$T(n) = 16T(n/16) + 4(n/4) + n$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 2

$T(n) = 4T(n/4) + n$ a = 4, b = 4, d = 1

$T(n) = 16(4T(n/64) + n/16)\ 4(n/4) + n$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 2

$T(n) = 4T(n/4) + n$ a = 4, b = 4, d = 1

$T(n) = 64T(n/64) + 16(n/16) + 4(n/4) + n$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 2

$T(n) = 4T(n/4) + n$ a = 4, b = 4, d = 1

$T(n) = 64T(n/64) + 16(n/16) + 4(n/4) + n$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

$\vdots$

(log$_4$ n times)

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 2

$T(n) = 4T(n/4) + n$ a = 4, b = 4, d = 1

$T(n) = 64T(n/64) + 16(n/16) + 4(n/4) + n$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

$\vdots$

(log$_4$ n times)

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 2

$T(n) = 4T(n/4) + n$ a = 4, b = 4, d = 1

$T(n) = 64T(n/64) + 16(n/16) + 4(n/4) + n$

$T(n) \in \Theta(n \log n)$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 3

$T(n) = T(n/2) + n$ $\qquad$ a = 1, b = 2, d = 1

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 3

$T(n) = T(n/2) + n$

$a = 1,\ b = 2,\ d = 1$

$a < b^d$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

So, by the Master Theorem, $T(n) \in \Theta(n)$

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 3

$T(n) = T(n/2) + n$ $\qquad$ $a = 1, b = 2, d = 1$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 3

$T(n) = T(n/2) + n$ a = 1, b = 2, d = 1

$T(n) = T(n/4) + n/2 + n$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 3

$T(n) = T(n/2) + n$ a = 1, b = 2, d = 1

$T(n) = T(n/8) + n/4 + n/2 + n$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 3

$T(n) = T(n/2) + n$ a = 1, b = 2, d = 1

$T(n) = T(n/8) + n/4 + n/2 + n$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 3

$T(n) = T(n/2) + n$ a = 1, b = 2, d = 1

$T(n) = T(n/8) + n/4 + n/2 + n$

$T(n) \in \Theta(n)$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 4

$T(n) = 2T(n/2) + n^2$ a = 2, b = 2, d = 2

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 4

$T(n) = 2T(n/2) + n^2$

$a = 2, b = 2, d = 2$

$a < b^d$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

So, by the Master Theorem, $T(n) \in \Theta(n^2)$

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 4

$T(n) = 2T(n/2) + n^2$ a = 2, b = 2, d = 2

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 4

$T(n) = 2T(n/2) + n^2$ a = 2, b = 2, d = 2

$T(n) = 2(2T(n/4) + (n/2)^2) + n^2$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 4

$T(n) = 2T(n/2) + n^2$ a = 2, b = 2, d = 2

$T(n) = 4T(n/4) + 2(n/2)^2 + n^2$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 4

$T(n) = 2T(n/2) + n^2$ a = 2, b = 2, d = 2

$T(n) = 4(2T(n/8) + (n/4)^2) + 2(n/2)^2 + n^2$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 4

$T(n) = 2T(n/2) + n^2$ $\qquad$ $a = 2,\ b = 2,\ d = 2$

$T(n) = 8T(n/8) + 4(n/4)^2 + 2(n/2)^2 + n^2$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 4

$T(n) = 2T(n/2) + n^2$ a = 2, b = 2, d = 2

$T(n) = 8T(n/8) + 4(n/4)^2 + 2(n/2)^2 + n^2$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Master Theorem: Example 4

$T(n) = 2T(n/2) + n^2$ a = 2, b = 2, d = 2

$T(n) = 8T(n/8) + 4(n/4)^2 + 2(n/2)^2 + n^2$

$T(n) \in \Theta(n^2)$

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort

- Perhaps the most obvious application of divide-and-conquer:
- To sort an array (or a list), cut it into two halves, sort each half, and merge the two results.

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Mergesort

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Mergesort

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Mergesort

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Mergesort

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Mergesort

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Mergesort

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Mergesort

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Mergesort

THE UNIVERSITY OF MELBOURNE

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Mergesort

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Mergesort

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Mergesort

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Mergesort

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Mergesort

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays

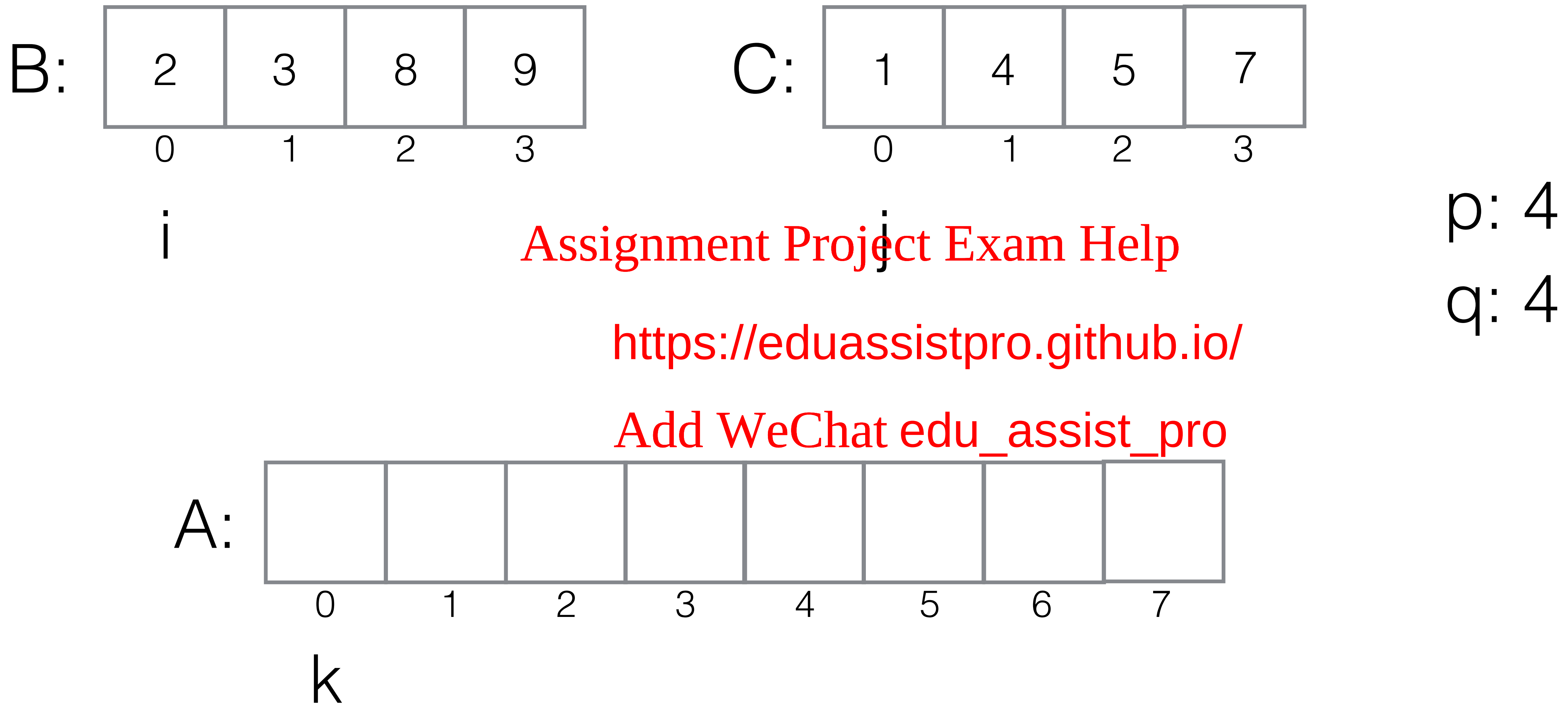
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:

2	3	8	9
0	1	2	3

C:

1	4	5	7
0	1	2	3

p: 4

q: 4

A:

0	1	2	3	4	5	6	7

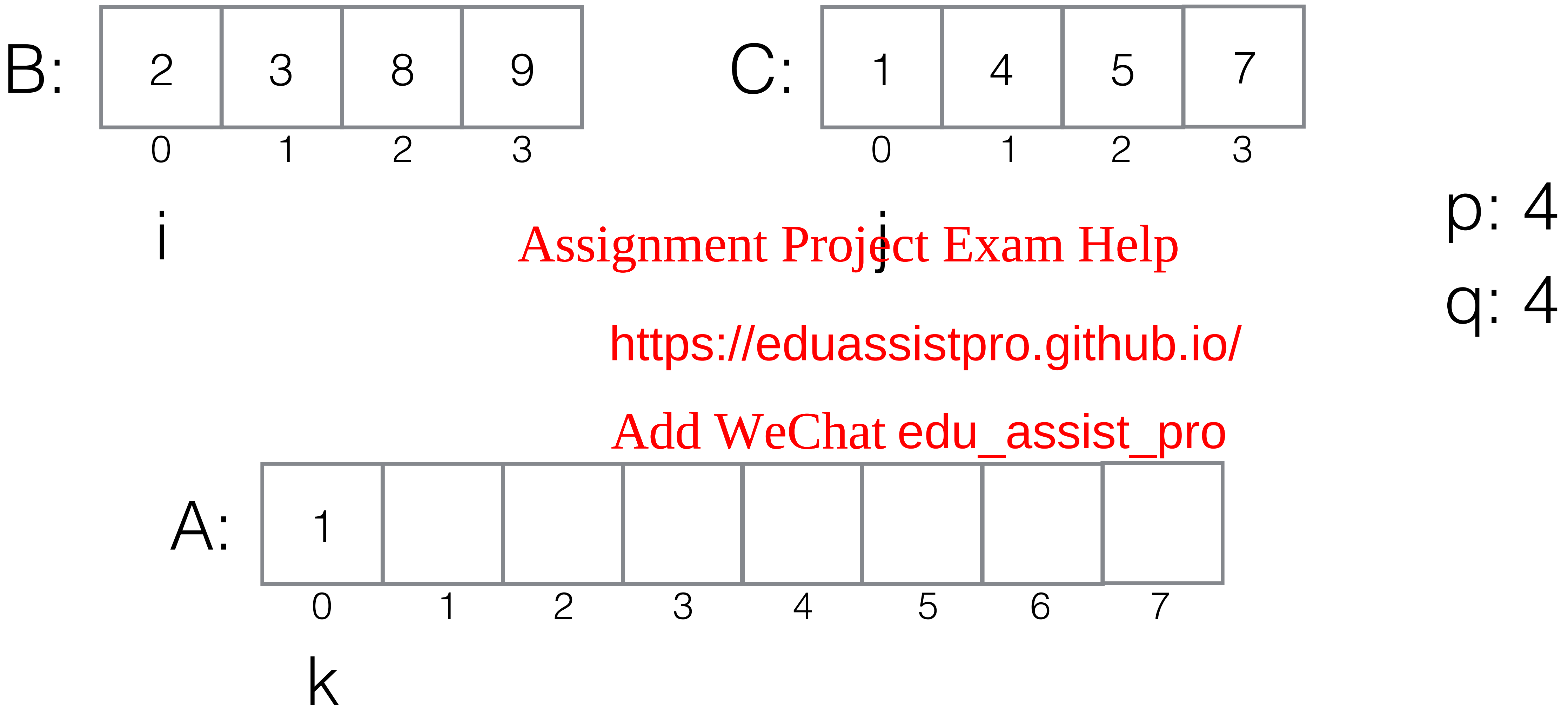
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:

2	3	8	9
0	1	2	3

i

C:

1	4	5	7
0	1	2	3

j

p: 4

q: 4

A:

1							
0	1	2	3	4	5	6	7

k

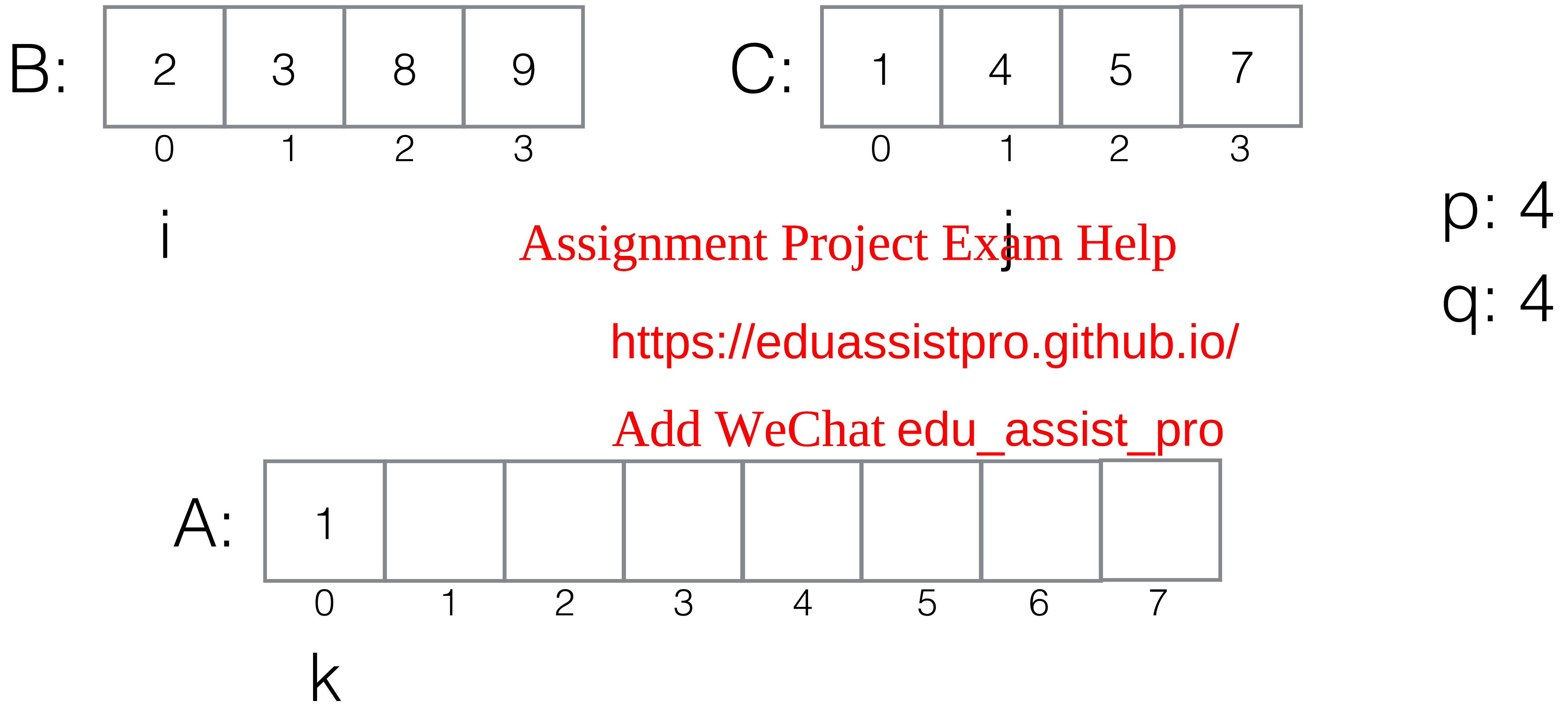
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:

0	1	2	3
2	3	8	9

C:

0	1	2	3
1	4	5	7

A:

0	1	2	3	4	5	6	7
1							

p: 4

q: 4

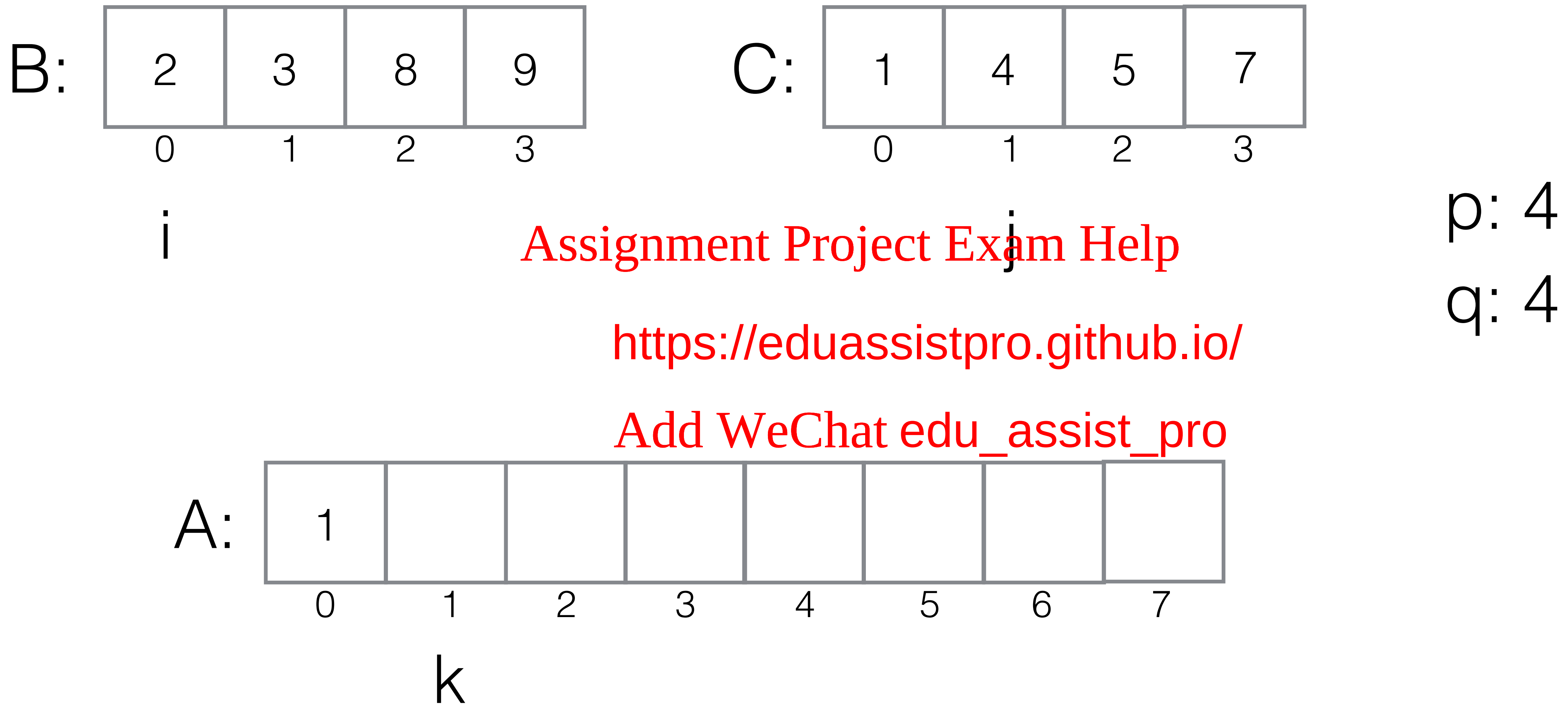
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:

0	1	2	3
2	3	8	9

i

C:

0	1	2	3
1	4	5	7

j

p: 4

q: 4

A:

0	1	2	3	4	5	6	7
1							

k

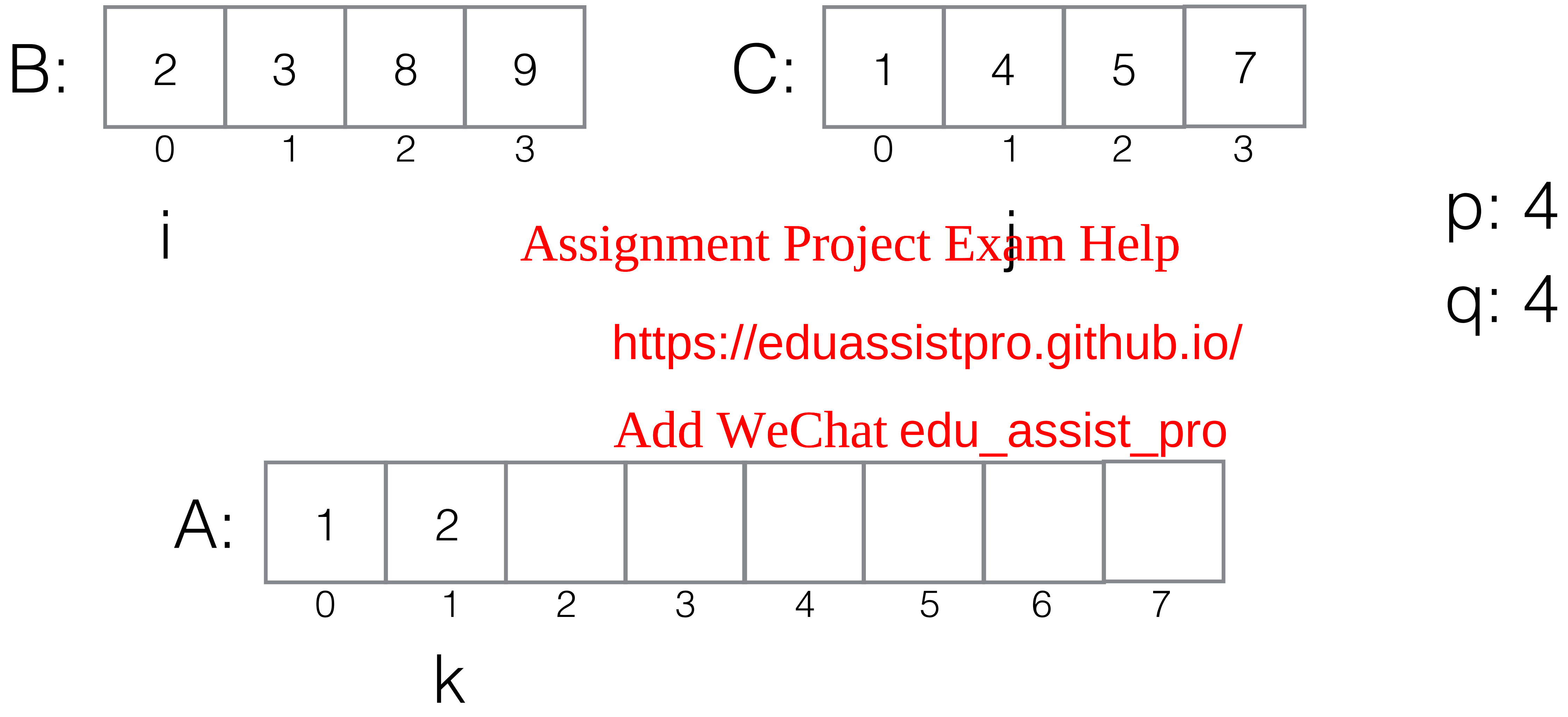
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:

2	3	8	9
0	1	2	3

i

C:

1	4	5	7
0	1	2	3

j

p: 4

q: 4

A:

1	2						
0	1	2	3	4	5	6	7

k

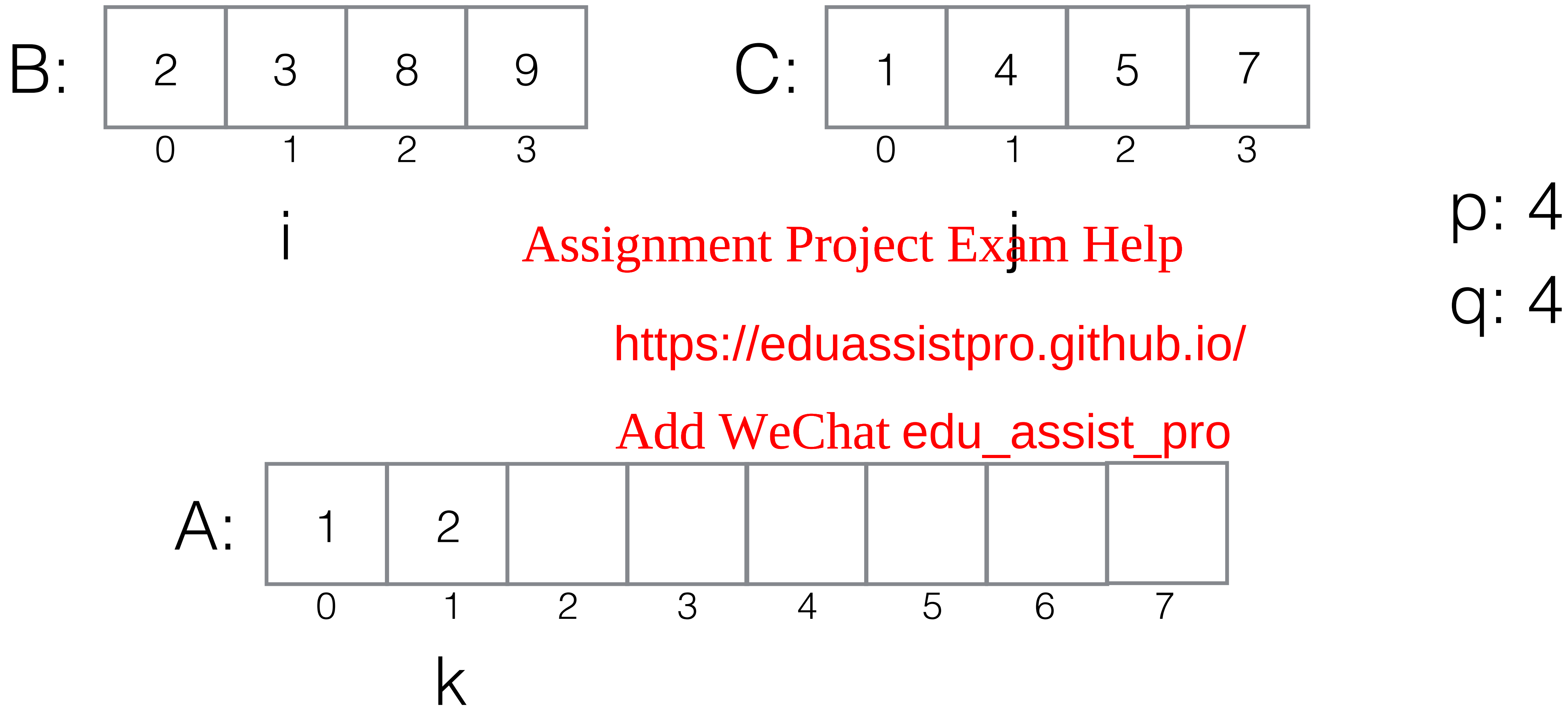
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:

2	3	8	9
0	1	2	3
	i		

C:

1	4	5	7
0	1	2	3
	j		

p: 4

q: 4

A:

1	2						
0	1	2	3	4	5	6	7
	k						

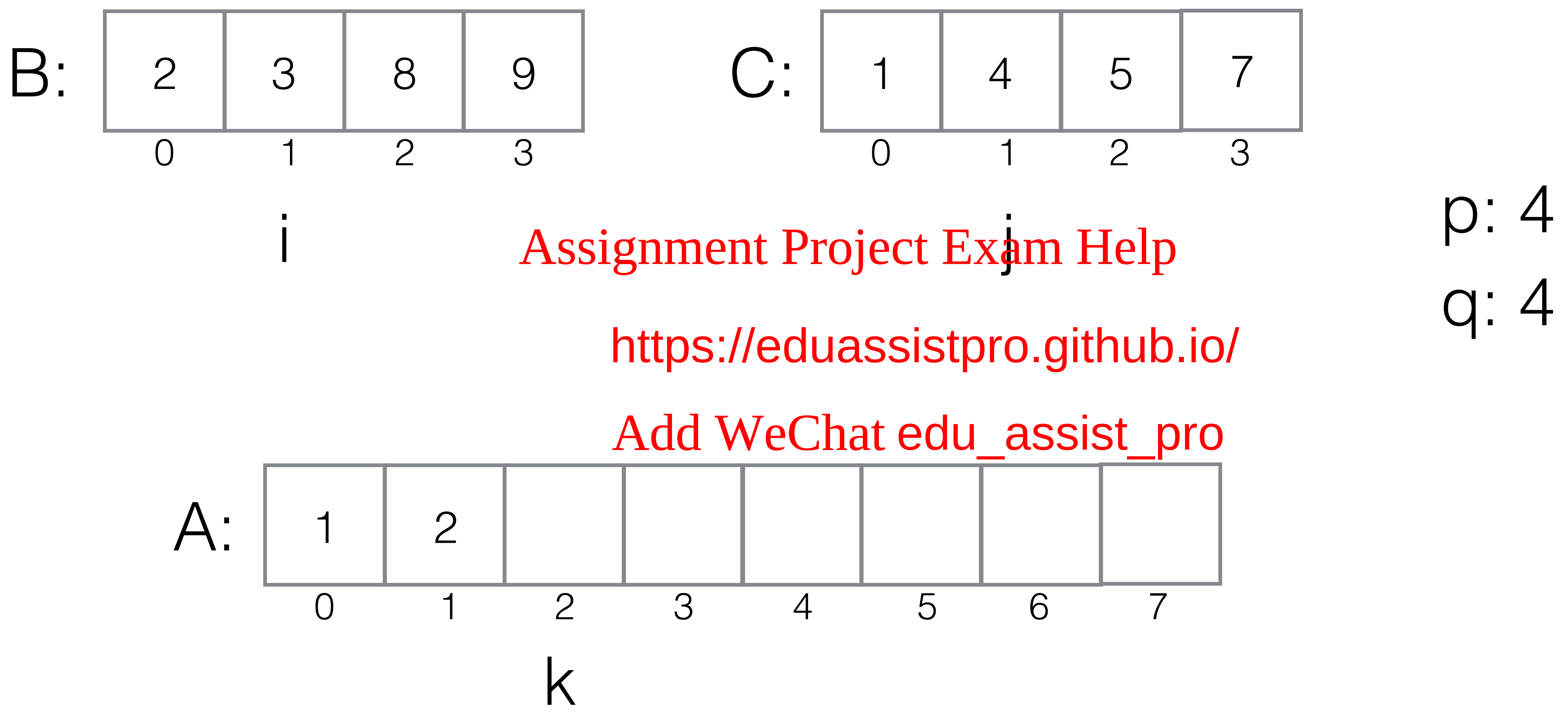
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:	2	3	8	9
	0	1	2	3
		i		

C:	1	4	5	7
	0	1	2	3
		j		

p: 4

q: 4

A:	1	2						
	0	1	2	3	4	5	6	7
			k					

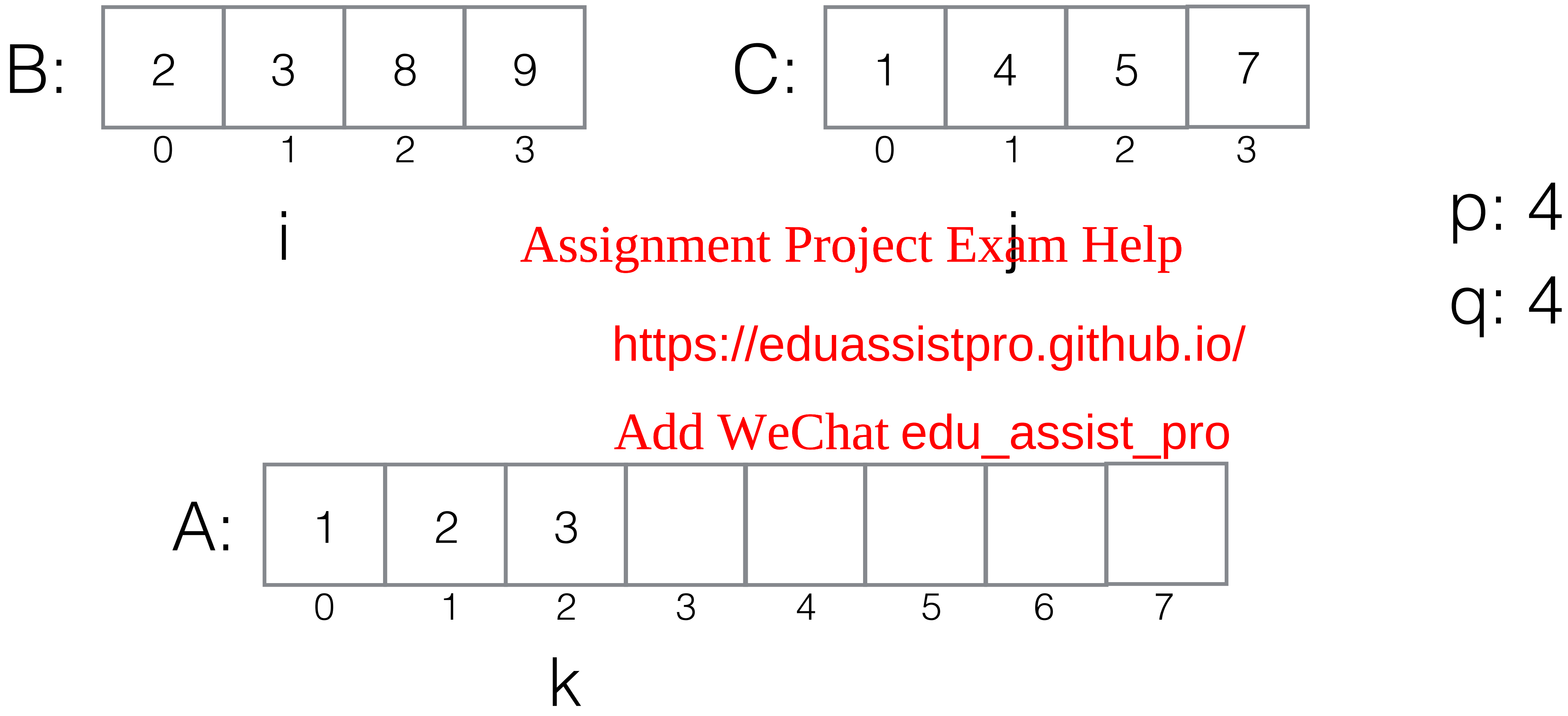
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:

2	3	8	9
0	1	2	3

i

C:

1	4	5	7
0	1	2	3

j

p: 4

q: 4

A:

1	2	3					
0	1	2	3	4	5	6	7

k

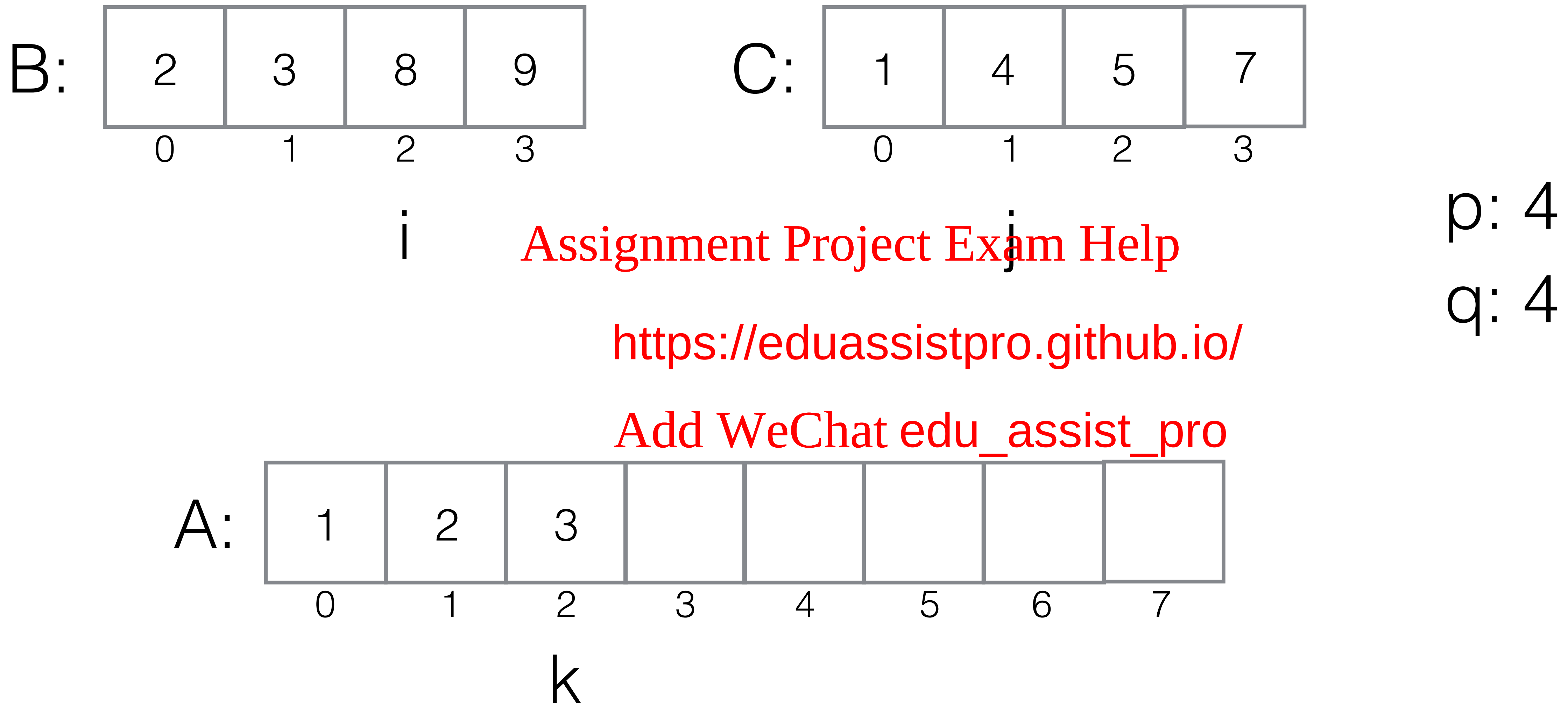
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:

2	3	8	9
0	1	2	3

i

C:

1	4	5	7
0	1	2	3

j

p: 4

q: 4

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

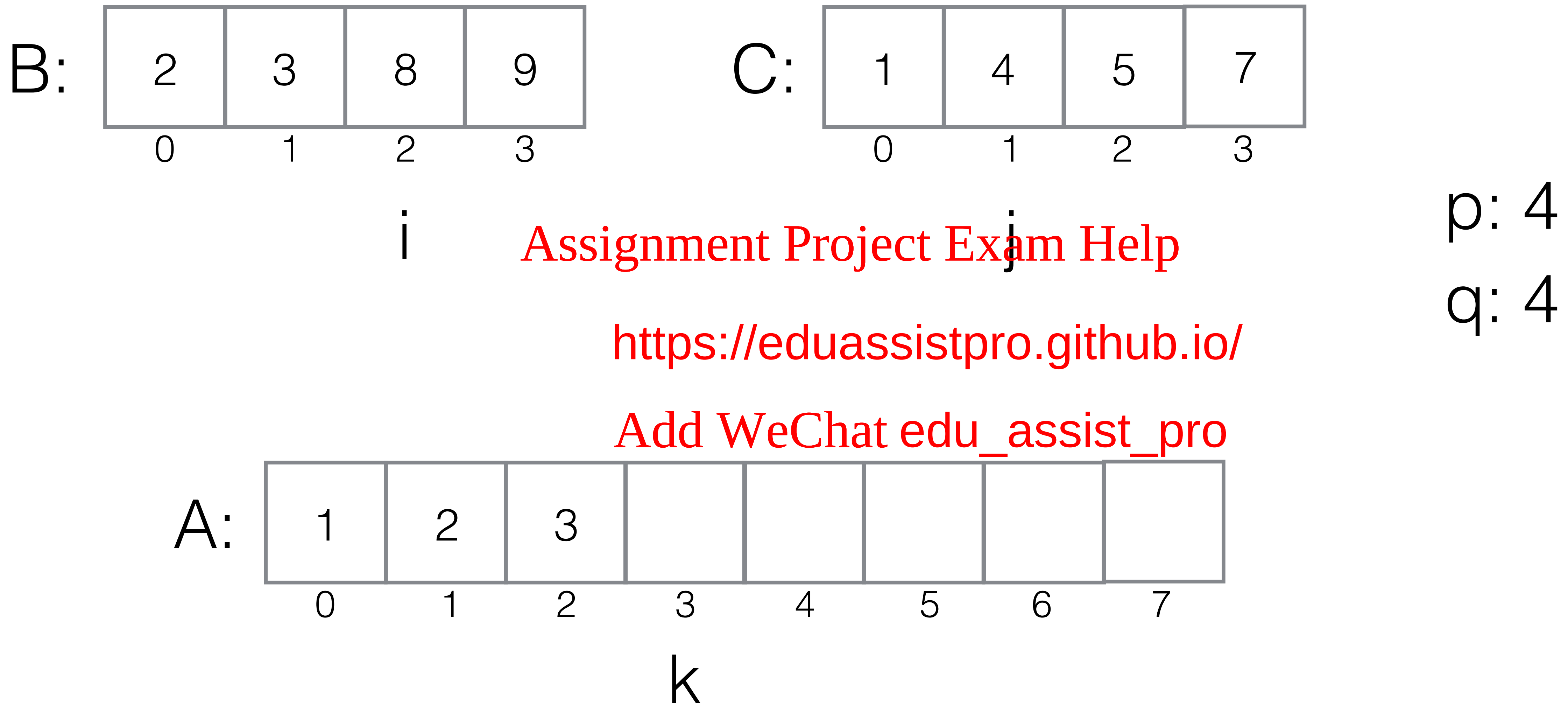
A:

1	2	3					
0	1	2	3	4	5	6	7

k

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:

0	1	2	3
2	3	8	9

C:

0	1	2	3
1	4	5	7

p: 4

q: 4

Assignment Project Exam Help

https://eduassistpro.github.io/

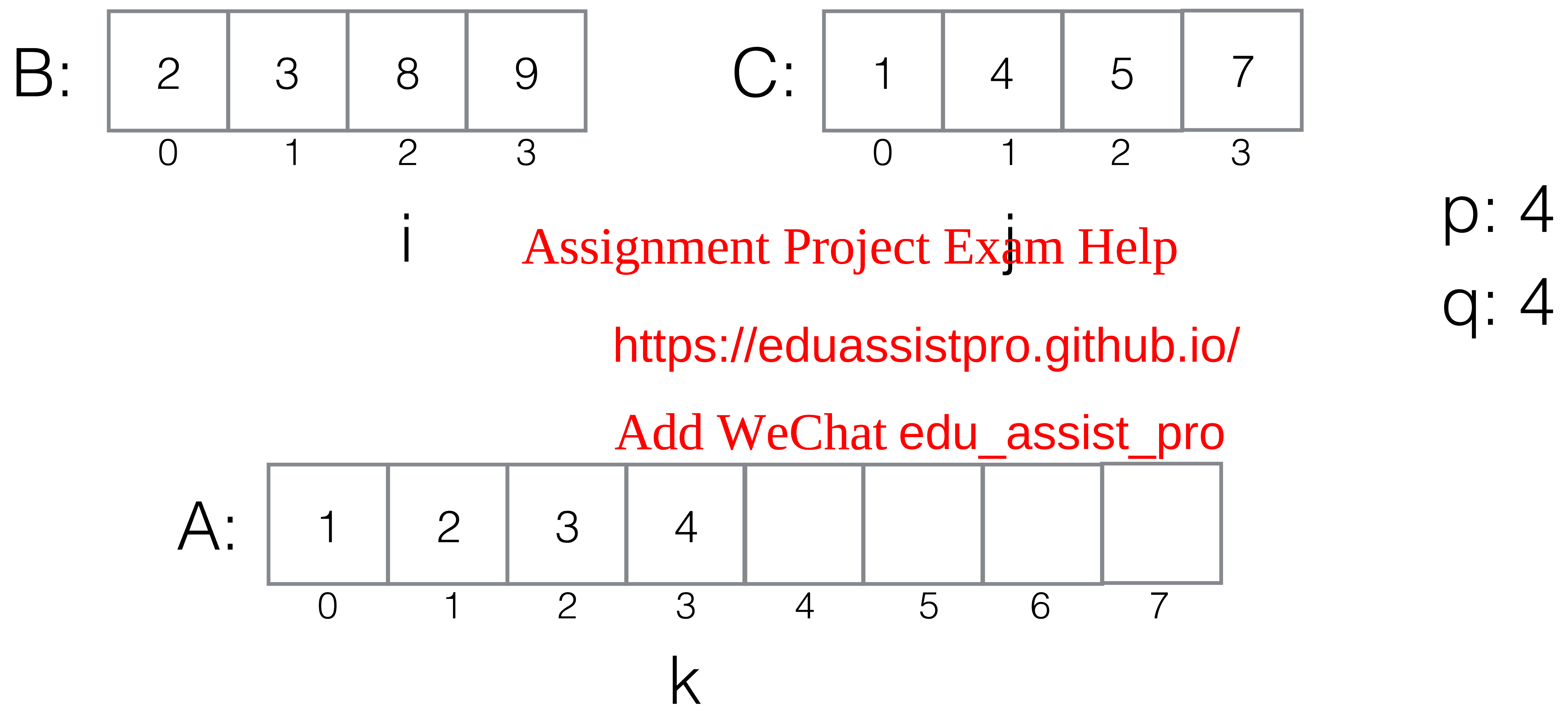
Add WeChat edu_assist_pro

A:

0	1	2	3	4	5	6	7
1	2	3					

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:

2	3	8	9
0	1	2	3
		i	

C:

1	4	5	7
0	1	2	3
	j		

p: 4

q: 4

A:

1	2	3	4				
0	1	2	3	4	5	6	7
			k				

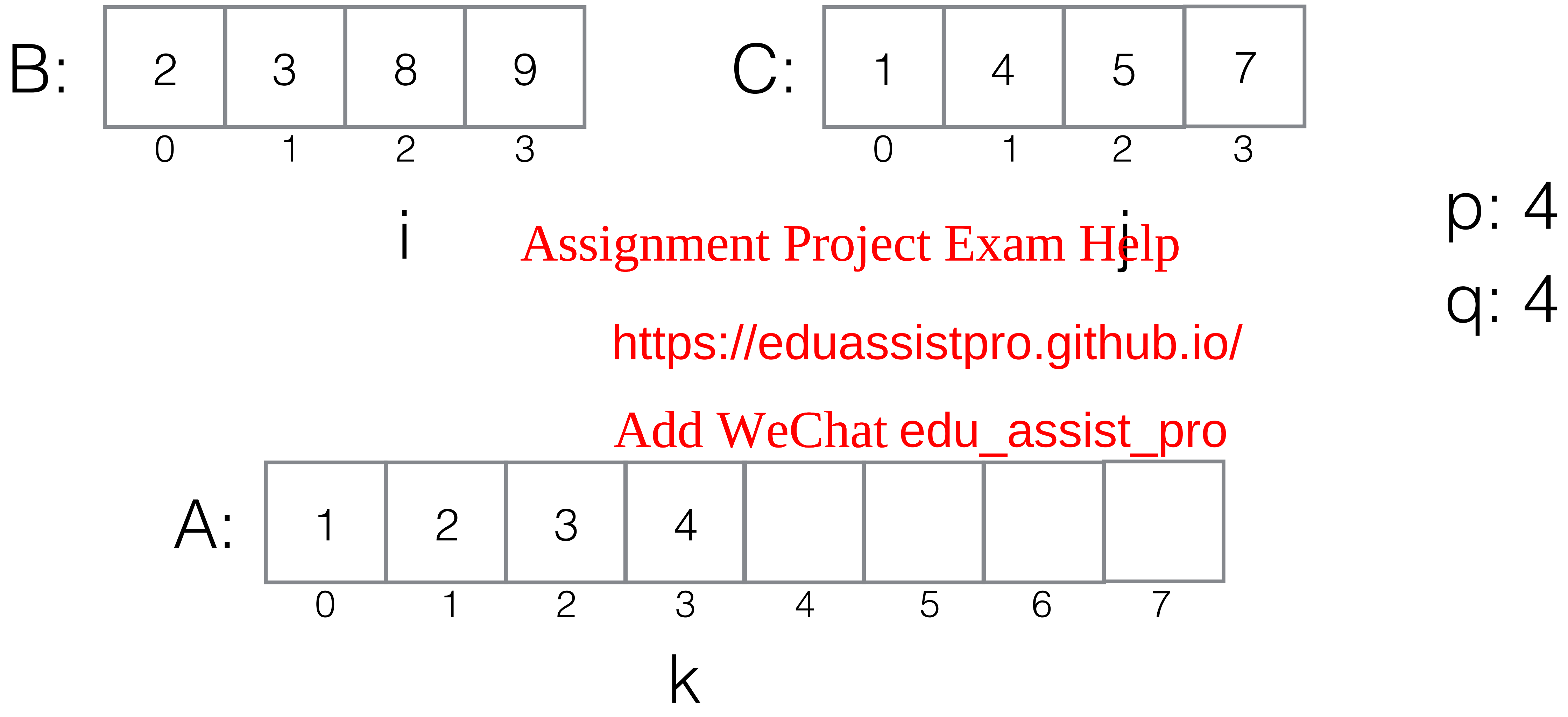
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:

2	3	8	9
0	1	2	3
		i	

C:

1	4	5	7
0	1	2	3
		j	

p: 4

q: 4

A:

1	2	3	4				
0	1	2	3	4	5	6	7
			k				

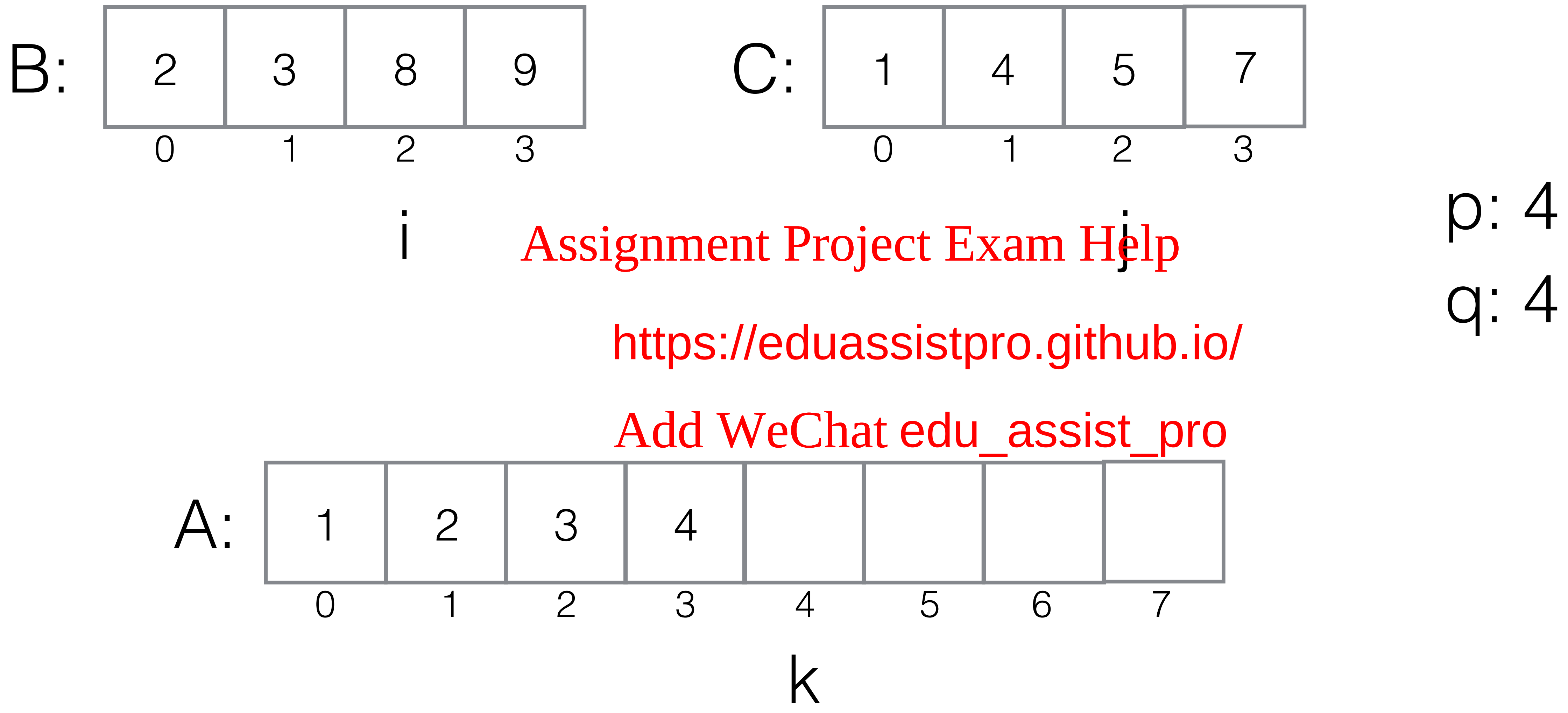
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:

2	3	8	9
0	1	2	3
		i	

C:

1	4	5	7
0	1	2	3
		j	

p: 4

q: 4

A:

1	2	3	4				
0	1	2	3	4	5	6	7
				k			

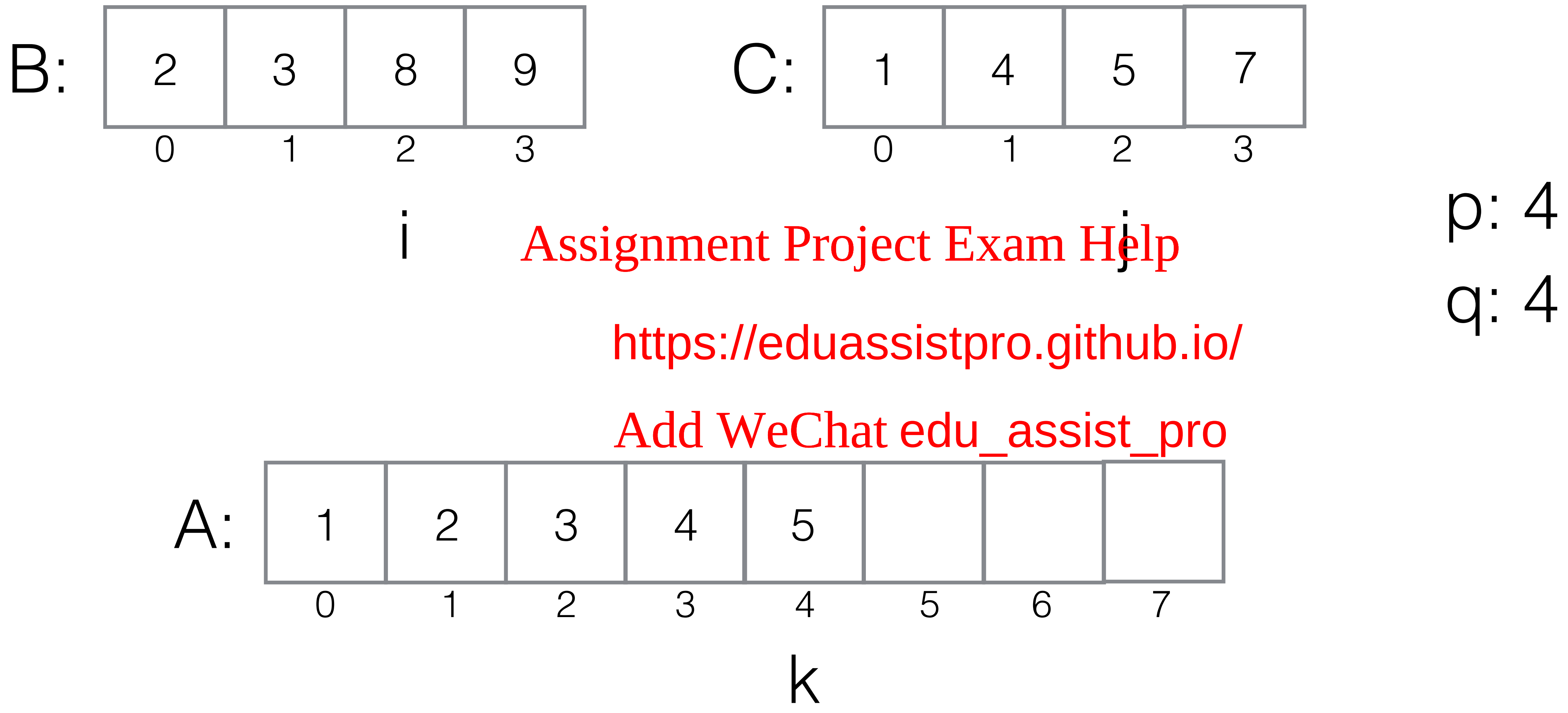
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:

2	3	8	9
0	1	2	3

i

C:

1	4	5	7
0	1	2	3

j

p: 4

q: 4

A:

1	2	3	4	5			
0	1	2	3	4	5	6	7

k

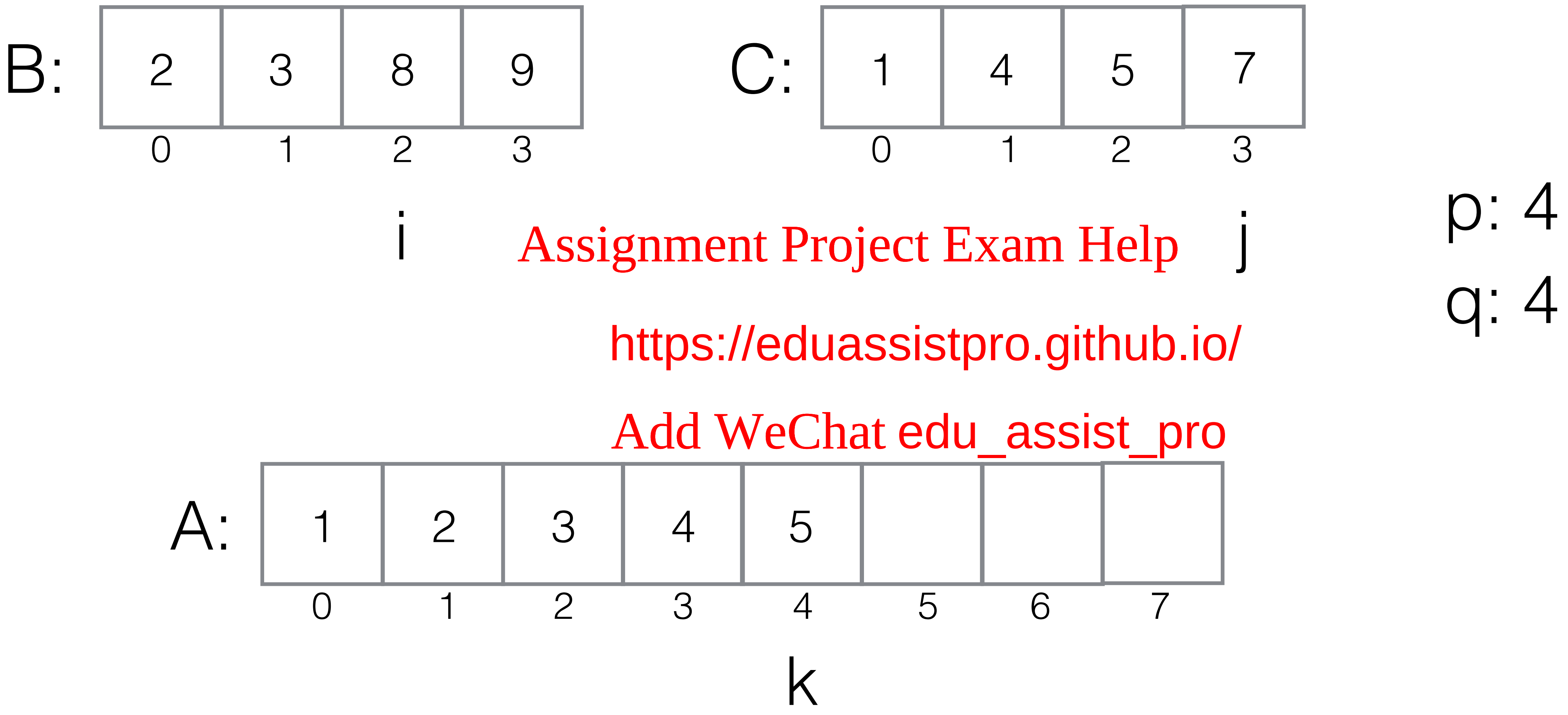
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:

2	3	8	9
0	1	2	3

C:

1	4	5	7
0	1	2	3

A:

1	2	3	4	5			
0	1	2	3	4	5	6	7

p: 4

q: 4

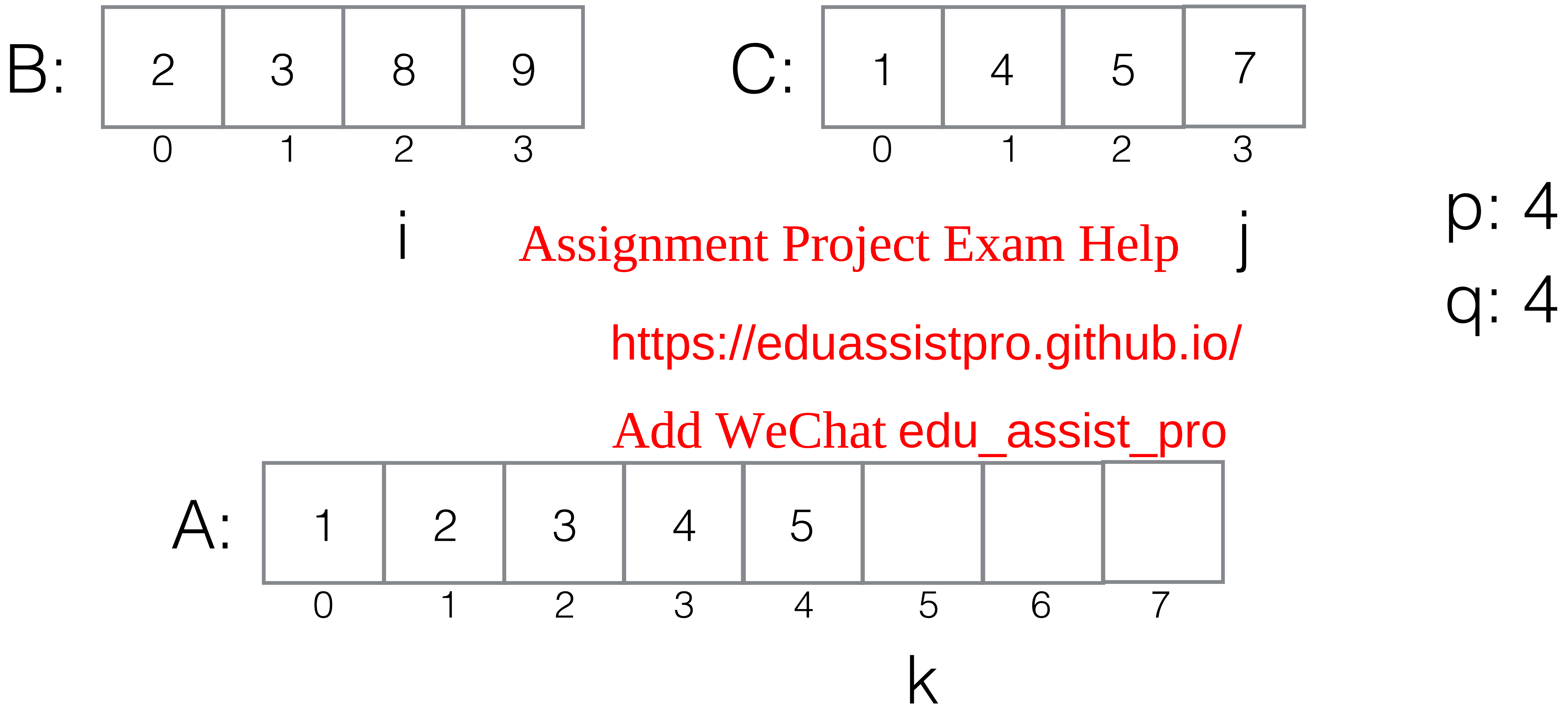
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:

0	1	2	3
2	3	8	9

C:

0	1	2	3
1	4	5	7

A:

0	1	2	3	4	5	6	7
1	2	3	4	5			

p: 4

q: 4

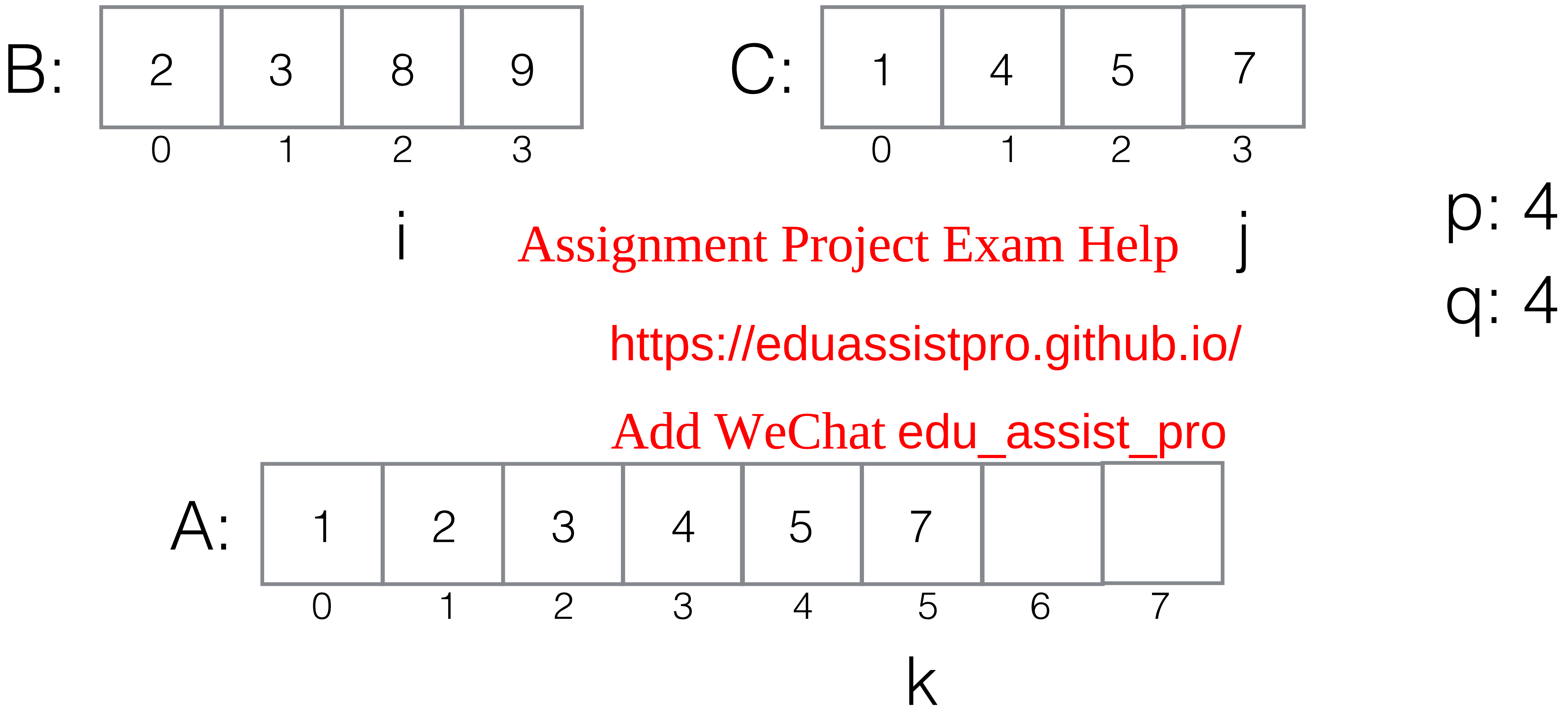
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:

2	3	8	9
0	1	2	3

i

C:

1	4	5	7
0	1	2	3

j

p: 4

q: 4

A:

1	2	3	4	5	7		
0	1	2	3	4	5	6	7

k

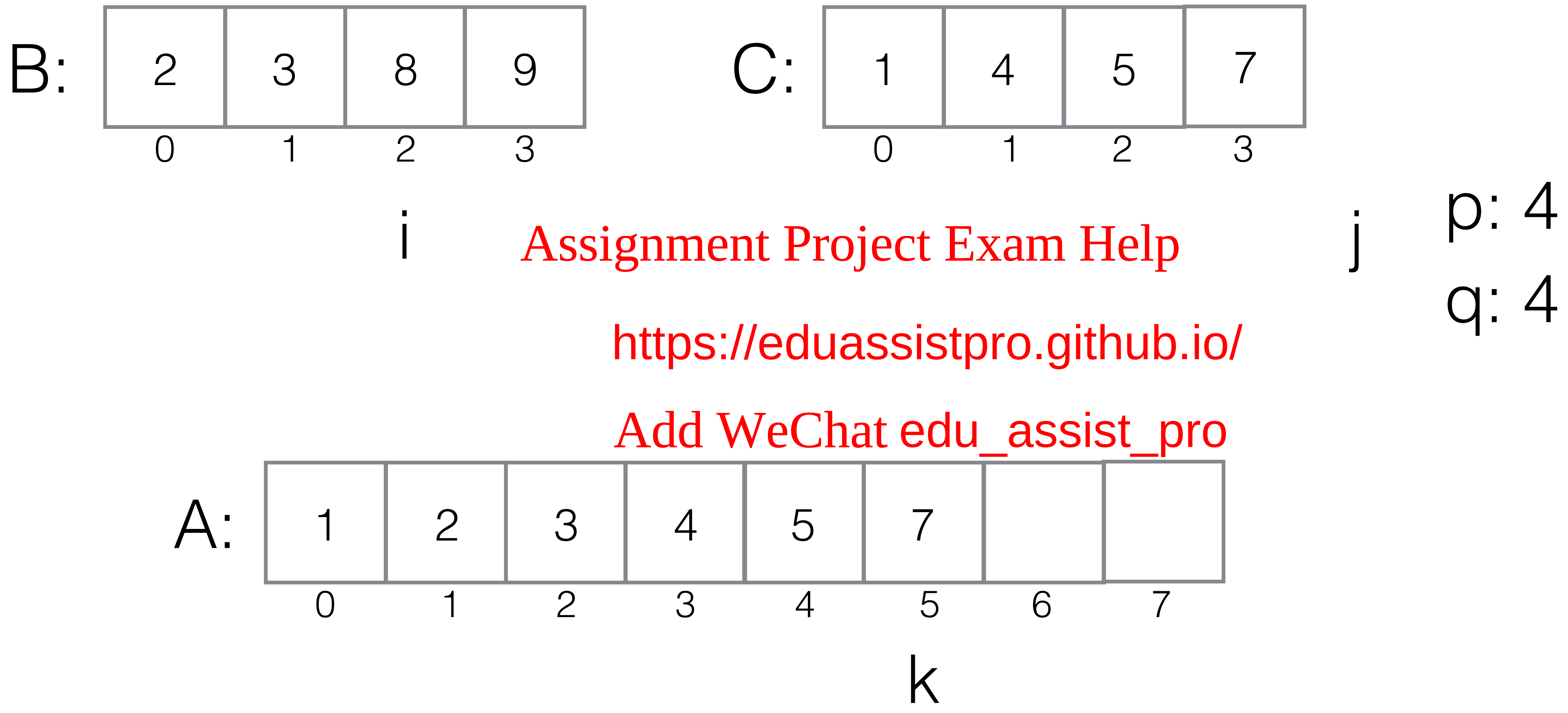
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:

2	3	8	9
0	1	2	3

C:

1	4	5	7
0	1	2	3

p: 4

q: 4

Assignment Project Exam Help

https://eduassistpro.github.io/

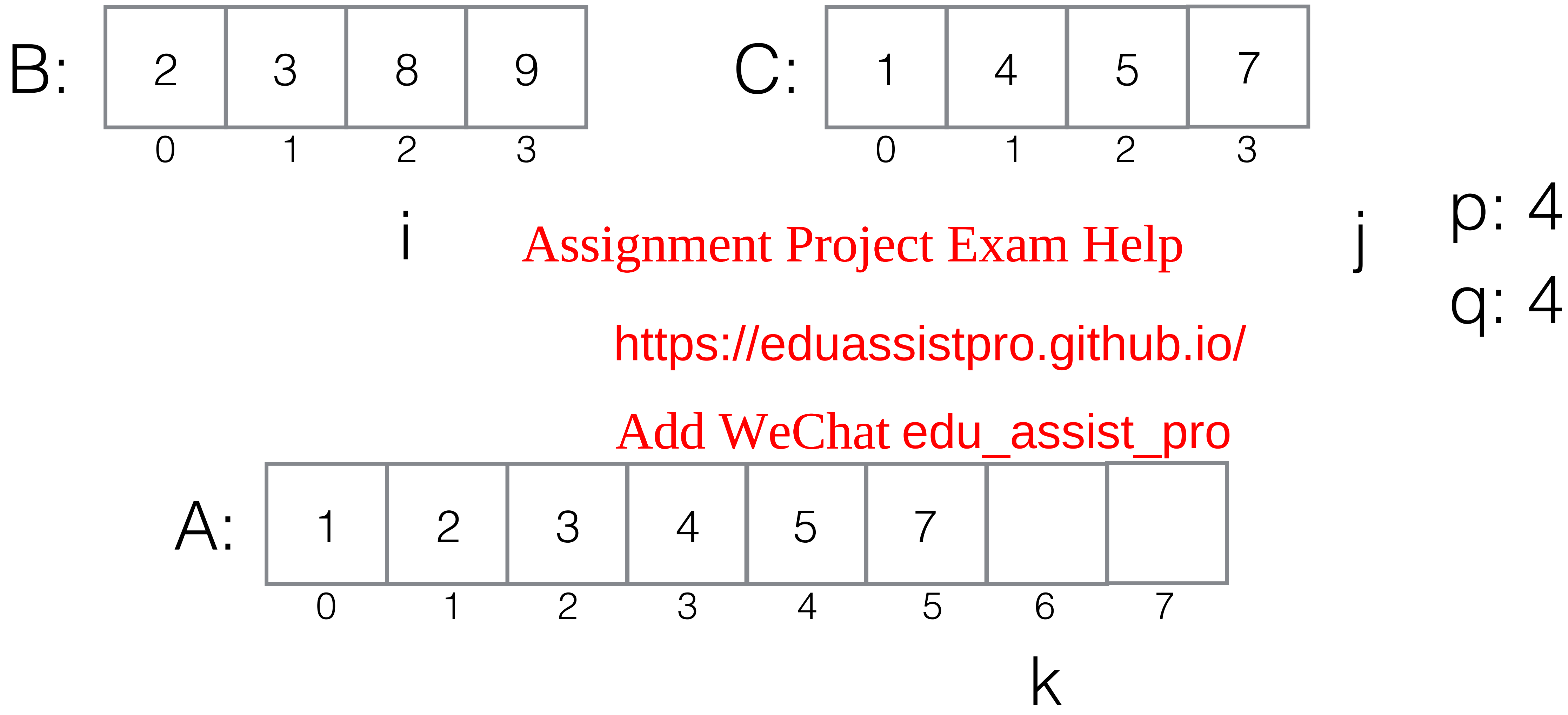
Add WeChat edu_assist_pro

A:

1	2	3	4	5	7		
0	1	2	3	4	5	6	7

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:

0	1	2	3
2	3	8	9

i

C:

0	1	2	3
1	4	5	7

j

p: 4

q: 4

A:

0	1	2	3	4	5	6	7
1	2	3	4	5	7		

k

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays

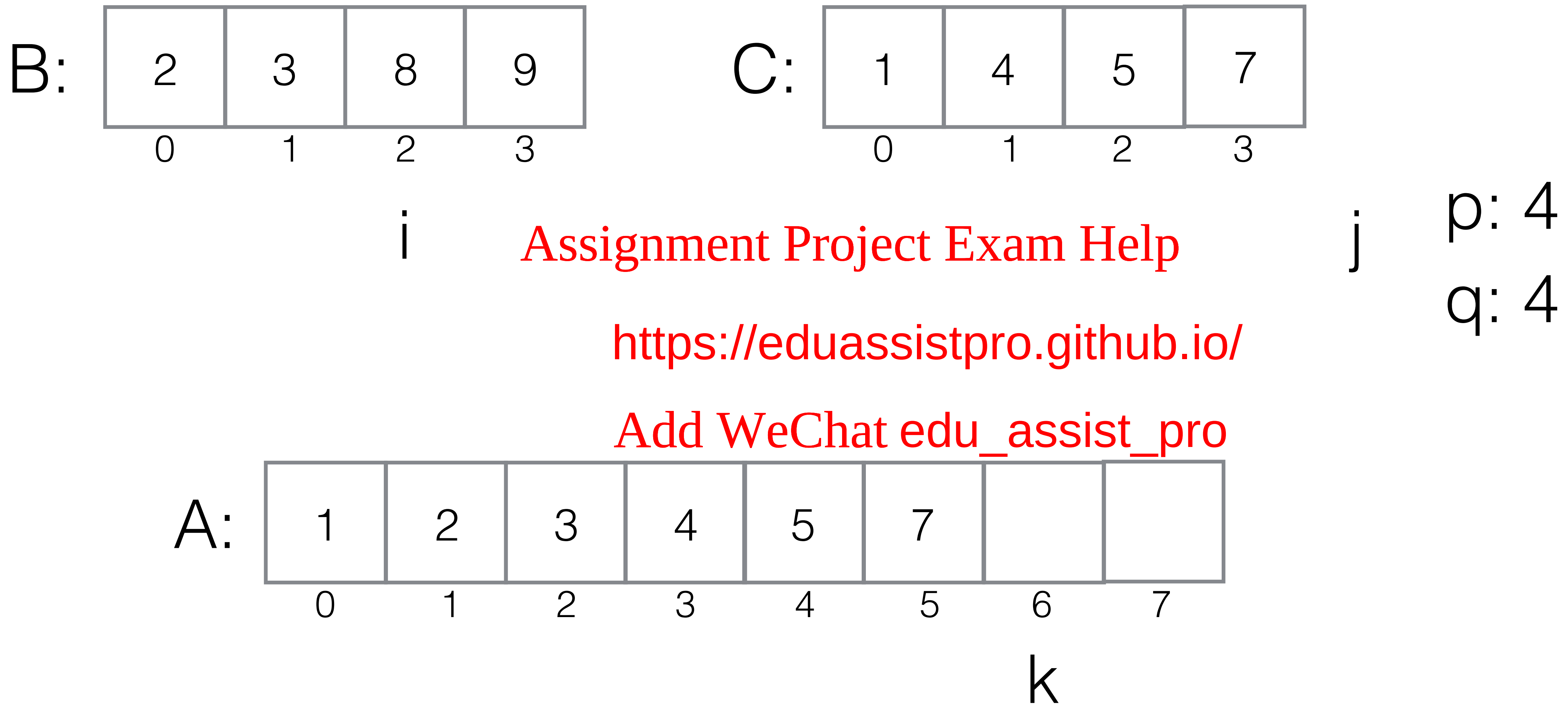
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:	2	3	8	9
	0	1	2	3

C:	1	4	5	7
	0	1	2	3

p: 4

q: 4

A:	1	2	3	4	5	7		
	0	1	2	3	4	5	6	7

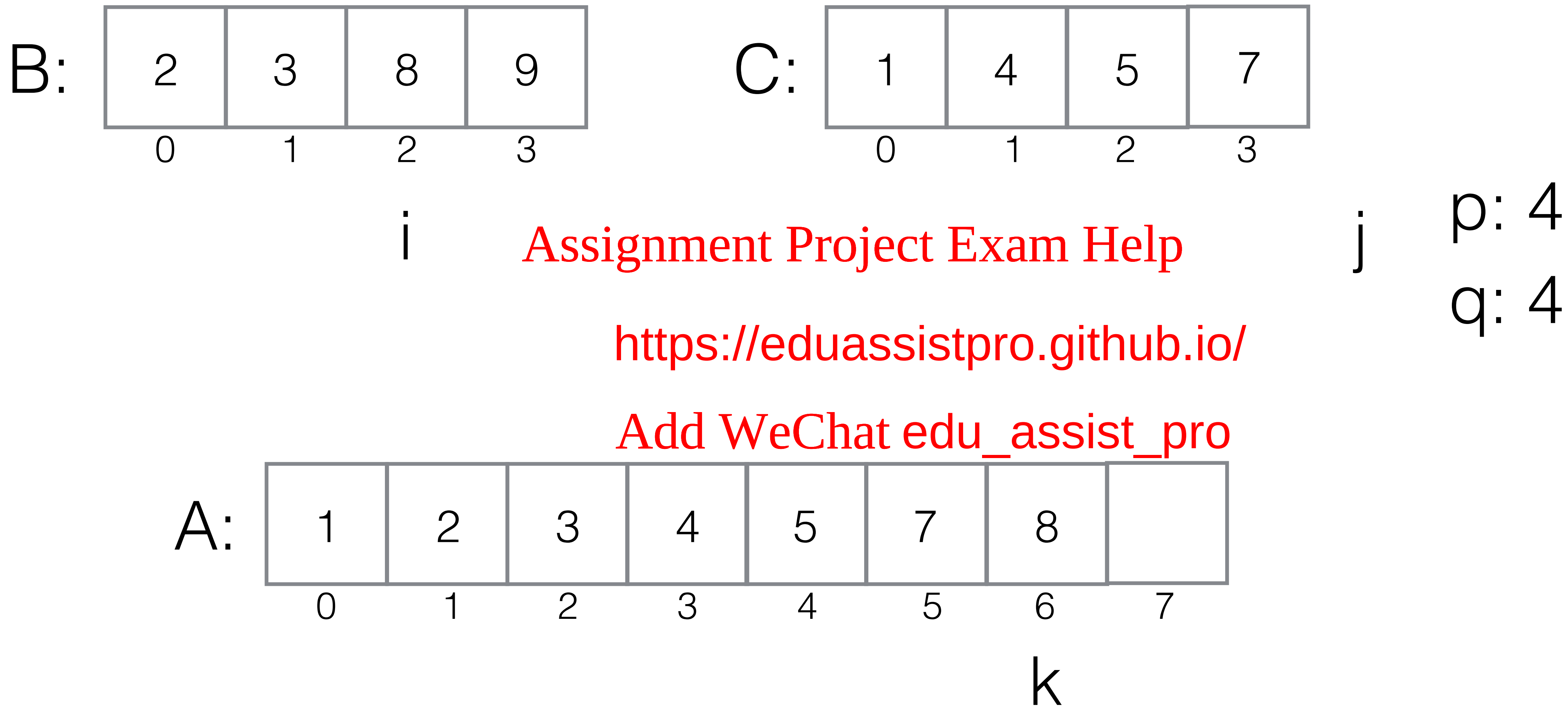
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:

2	3	8	9
0	1	2	3

C:

1	4	5	7
0	1	2	3

i

j

p: 4

q: 4

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

A:

1	2	3	4	5	7	8	
0	1	2	3	4	5	6	7

k

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays

B:

2	3	8	9
0	1	2	3

C:

1	4	5	7
0	1	2	3

i

j

p: 4

q: 4

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

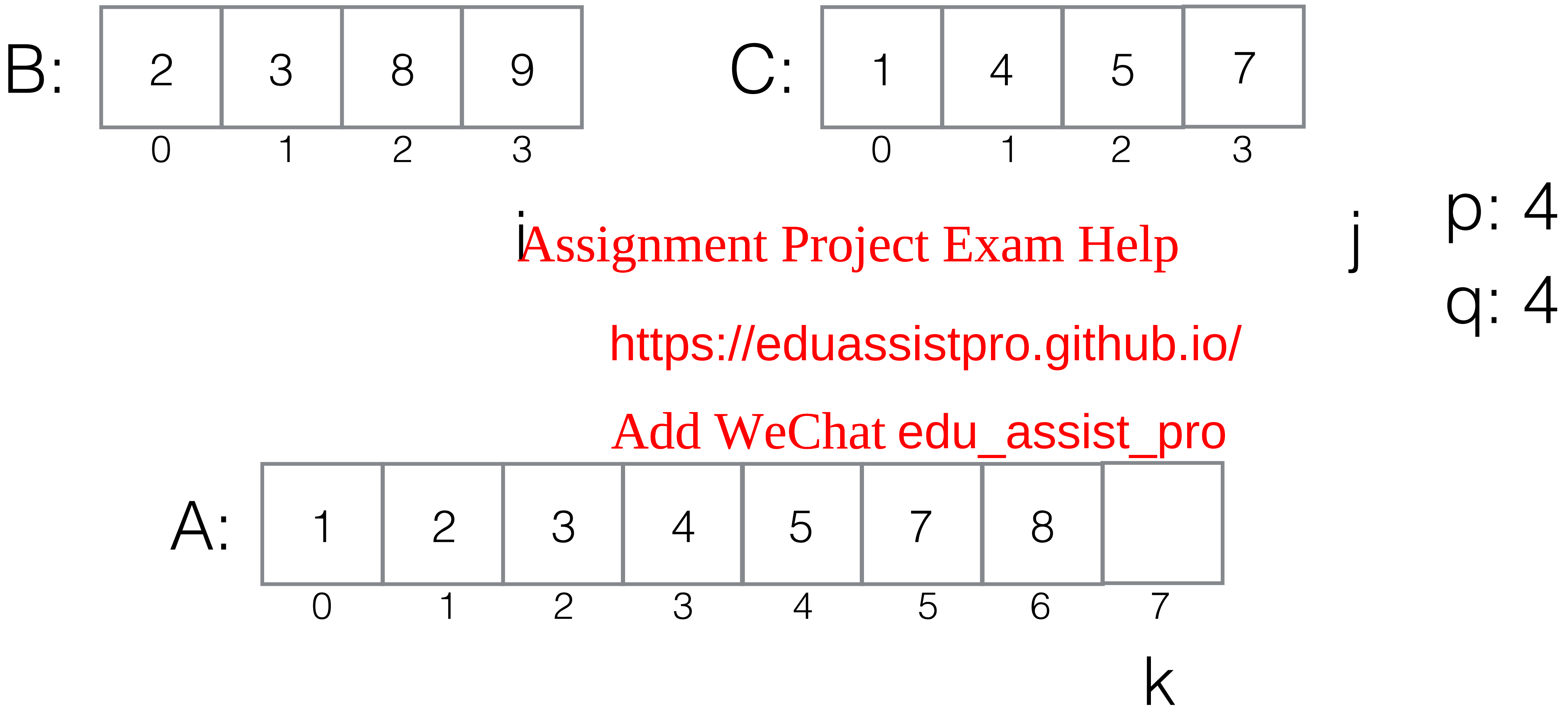
A:

1	2	3	4	5	7	8	
0	1	2	3	4	5	6	7

k

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:

0	1	2	3
2	3	8	9

C:

0	1	2	3
1	4	5	7

A:

0	1	2	3	4	5	6	7
1	2	3	4	5	7	8	

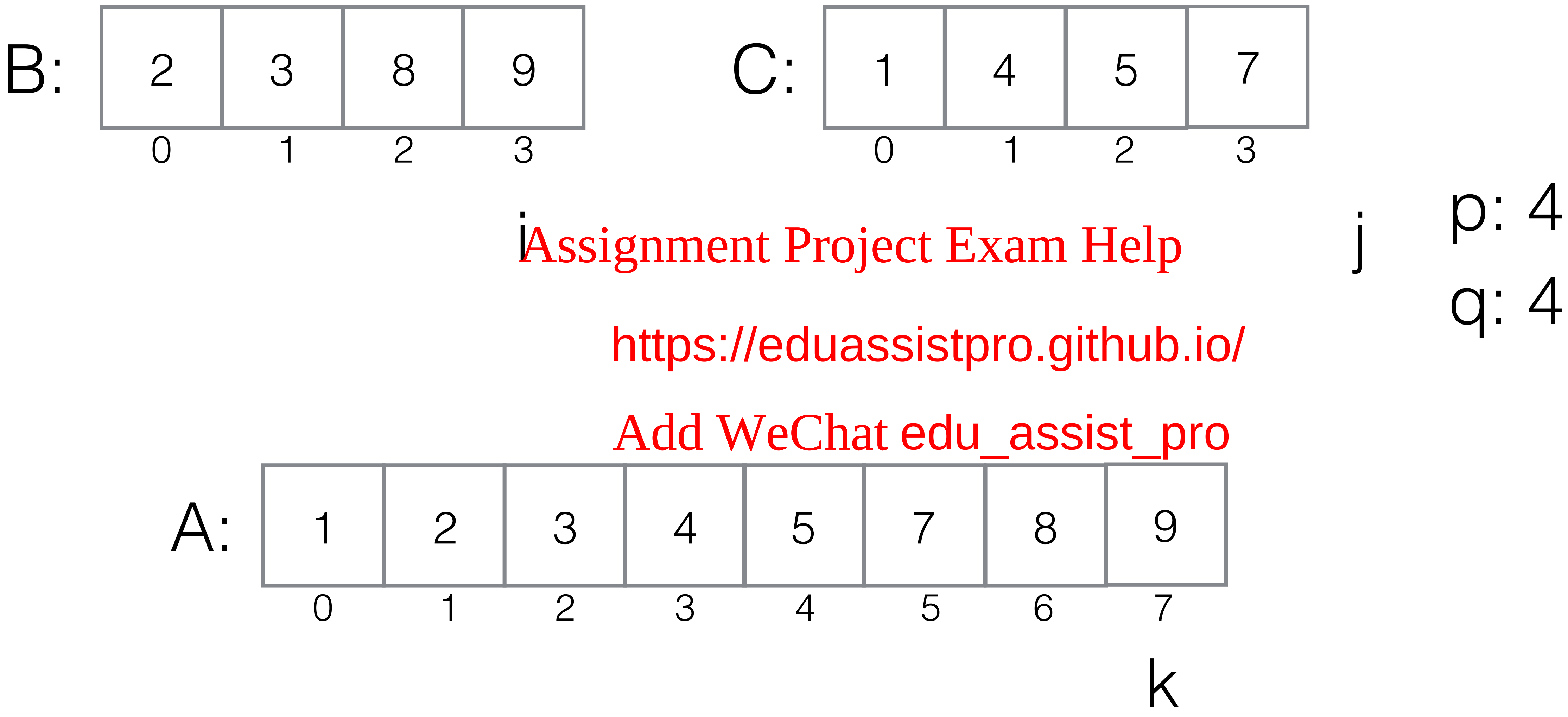
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Merging Arrays



B:

2	3	8	9
0	1	2	3

C:

1	4	5	7
0	1	2	3

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

A:

1	2	3	4	5	7	8	9
0	1	2	3	4	5	6	7

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Analysis

- How many comparisons will MERGE need to make in the worst case, when given arrays of size ⌊n/2⌋ and ⌈n/2⌉ ?

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

- If the largest and elements are in different arrays, then n − 1 risons. Hence the cost equation for Mergesor

$$C(n) = \begin{cases} 0 & \text{if } n < 2 \\ 2C(n/2) + n - 1 & \text{otherwise} \end{cases}$$

- By the Master Theorem, C(n) ∈ Θ(n log n).

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Properties

THE UNIVERSITY OF MELBOURNE

- For large n, the number of comparisons made tends to be around 75% of the worst-case scenario.
- Is mergesort stable? ?

Assignment Project Exam Help

- Is mergesort in-pl

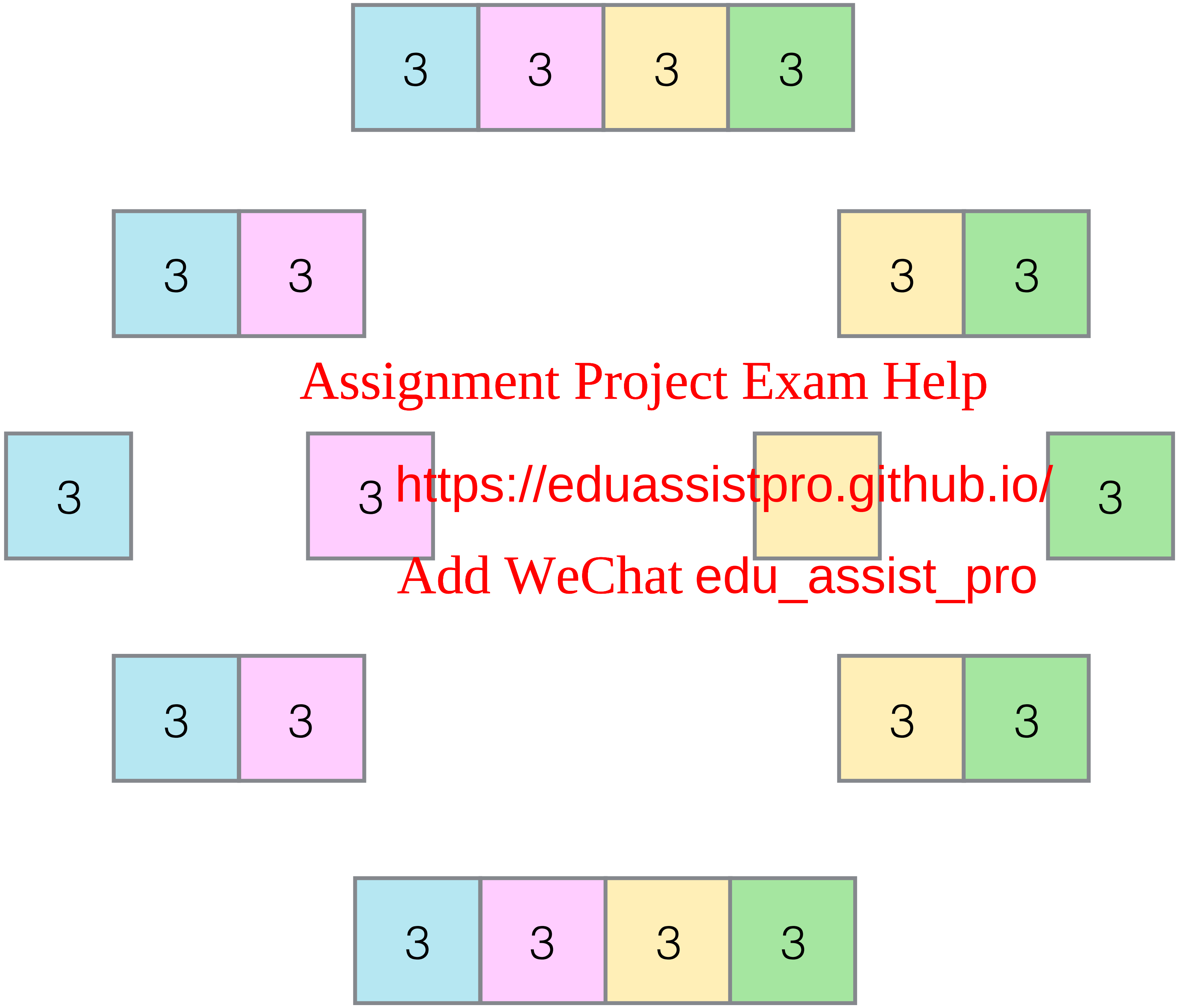
https://eduassistpro.github.io/

Add WeChat edu_assist_pro

- If comparisons are fast, me ranks between quicksort and heapsort (covered next week) for time, assuming random data.
- Mergesort is the method of choice for linked lists and for very large collections of data.

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Stability



Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Mergesort: Properties

- For large n, the number of comparisons made tends to be around 75% of the worst-case scenario.

- Is mergesort stable? yes

Assignment Project Exam Help

- Is mergesort in-pl

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

- If comparisons are fast, me ranks between quicksort and heapsort (covered next week) for time, assuming random data.

- Mergesort is the method of choice for linked lists and for very large collections of data.

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Bottom-Up Mergesort

- An alternative way of doing mergesort:

- Generate **runs** of length 2, then of length 4, and so on:

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort

- Quicksort takes a divide-and-conquer approach that is different to mergesort's.

- It uses the **partitioning** idea from QuickSelect, picking a pivot element, and partitioning the array around that, so as to obtain this situation:

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

- The element A[s] will be in its final position (it is the (s + 1)th smallest element).

- All that then needs to be done is to sort the segment to the left, recursively, as well as the segment to the right.

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort

- Very short and elegant:

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

- Initial call: Quicksort(A, 0, n – 1).

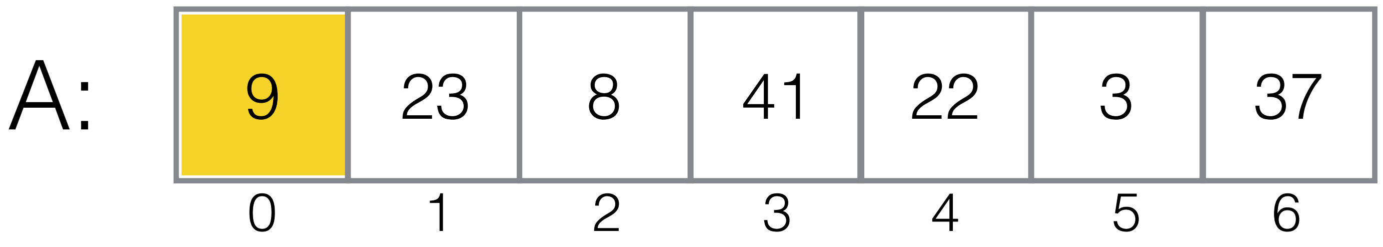
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort: Example

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



A:	0	1	2	3	4	5	6
	9	23	8	41	22	3	37

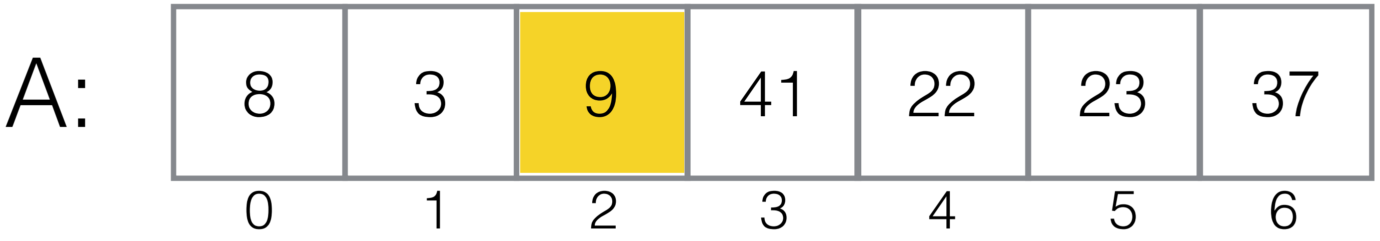
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort: Example

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



A:						
8	3	9	41	22	23	37
0	1	2	3	4	5	6

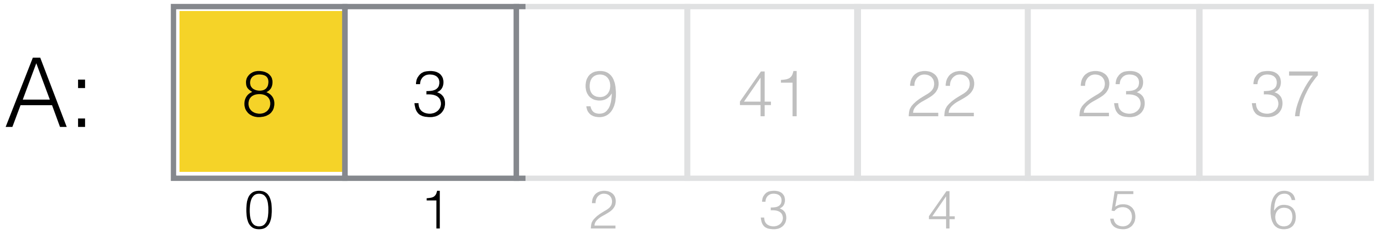
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort: Example

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



A:	8	3	9	41	22	23	37
	0	1	2	3	4	5	6

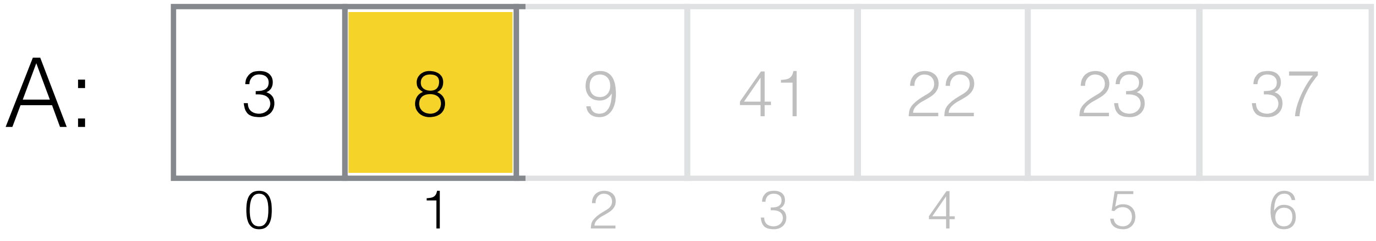
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort: Example

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



A:

3	8	9	41	22	23	37
0	1	2	3	4	5	6

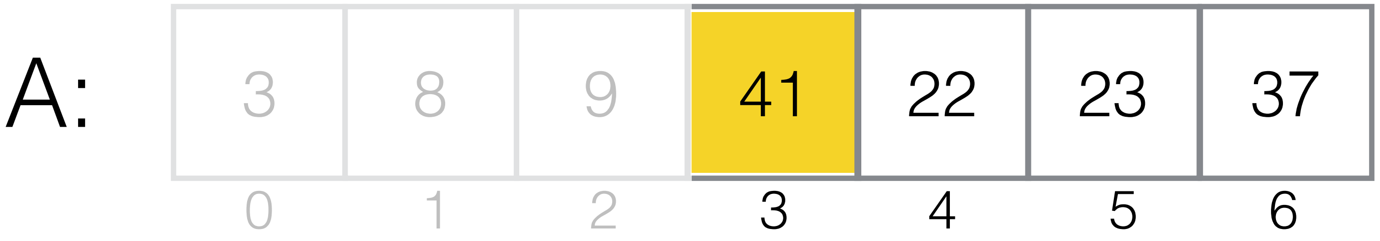
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort: Example

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



A:	3	8	9	41	22	23	37
	0	1	2	3	4	5	6

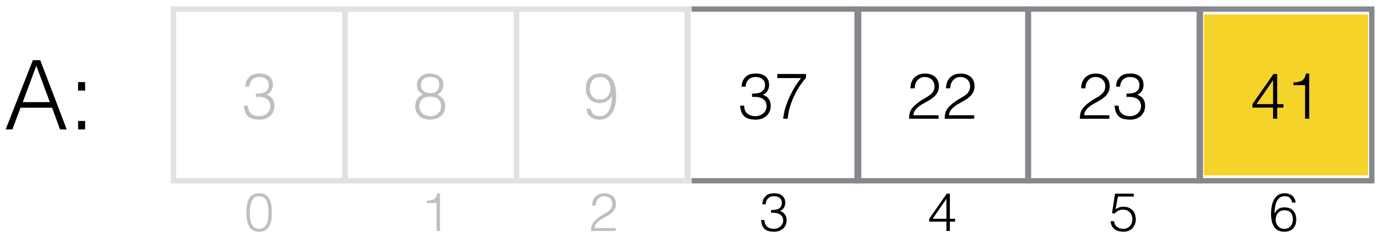
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort: Example

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



A:	3	8	9	37	22	23	41
	0	1	2	3	4	5	6

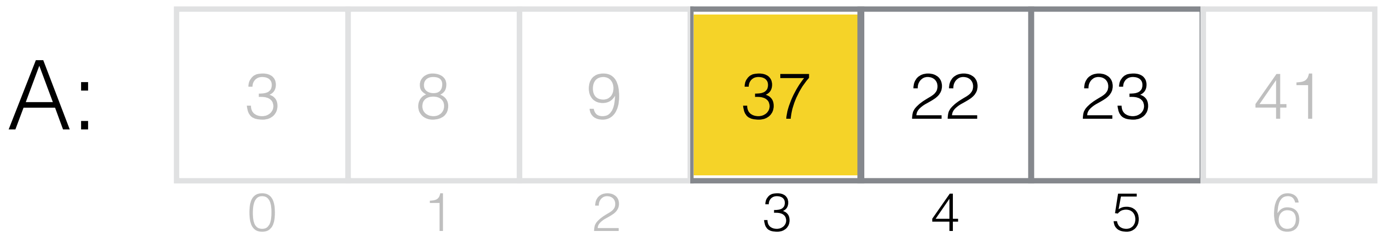
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort: Example

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



A:	3	8	9	37	22	23	41
	0	1	2	3	4	5	6

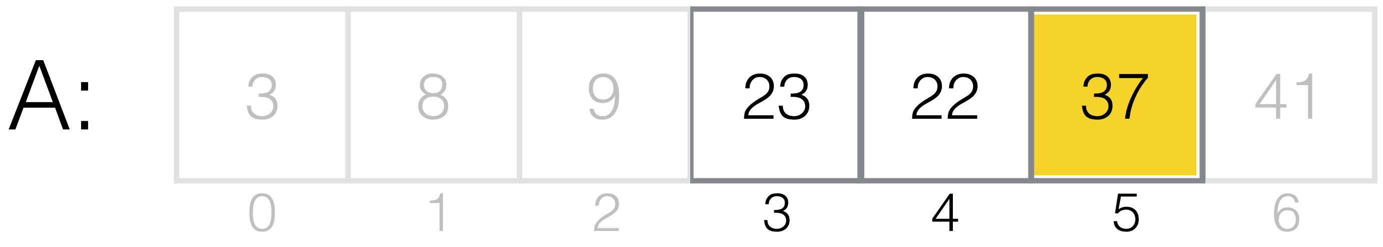
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort: Example

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



A:

3	8	9	23	22	37	41
0	1	2	3	4	5	6

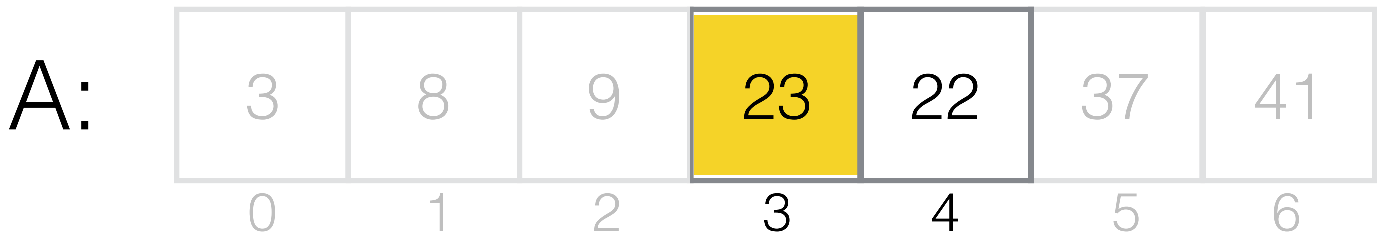
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort: Example

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



A:							
	3	8	9	23	22	37	41
	0	1	2	3	4	5	6

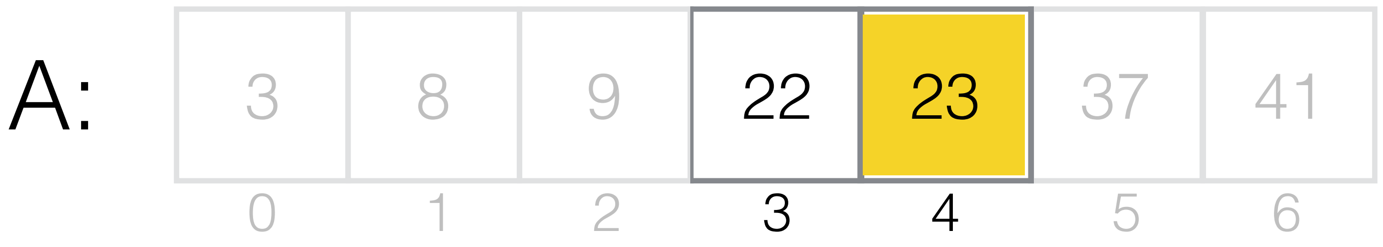
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort: Example

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



A:	3	8	9	22	23	37	41
	0	1	2	3	4	5	6

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort: Example

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

A:	3	8	9	22	23	37	41
	0	1	2	3	4	5	6

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Hoare Partitioning

- The standard way of doing partitioning in Quicksort

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Hoare Partitioning

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

A:

9	23	8	41	22	3	37
0	1	2	3	4	5	6
i						j

p: 9

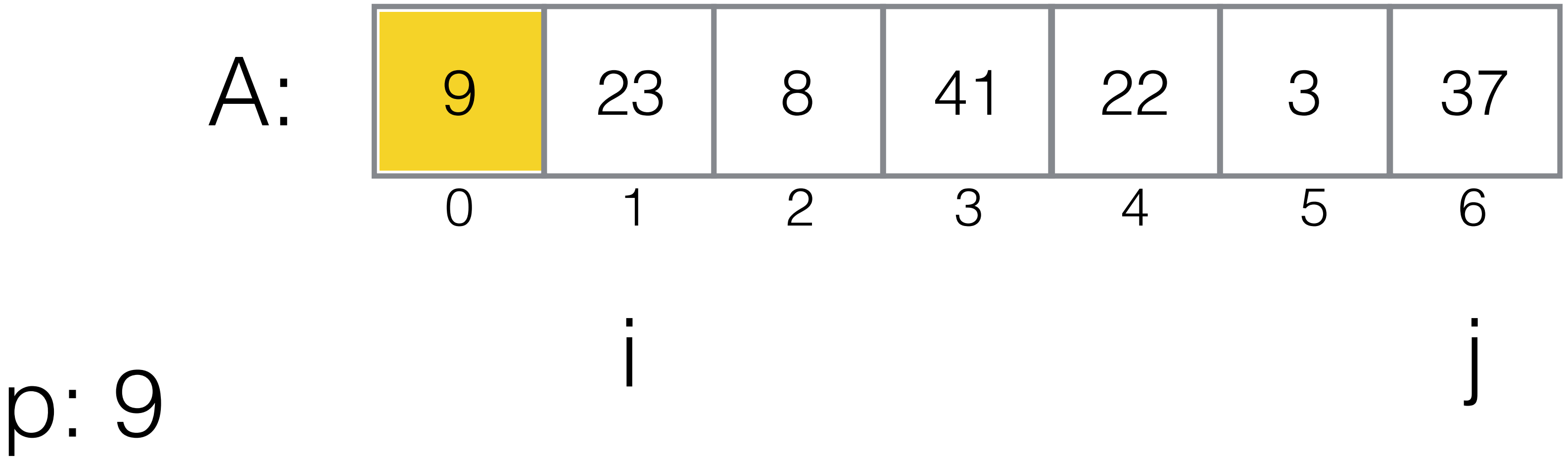
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Hoare Partitioning

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



A:	9	23	8	41	22	3	37
	0	1	2	3	4	5	6
		i					j

p: 9

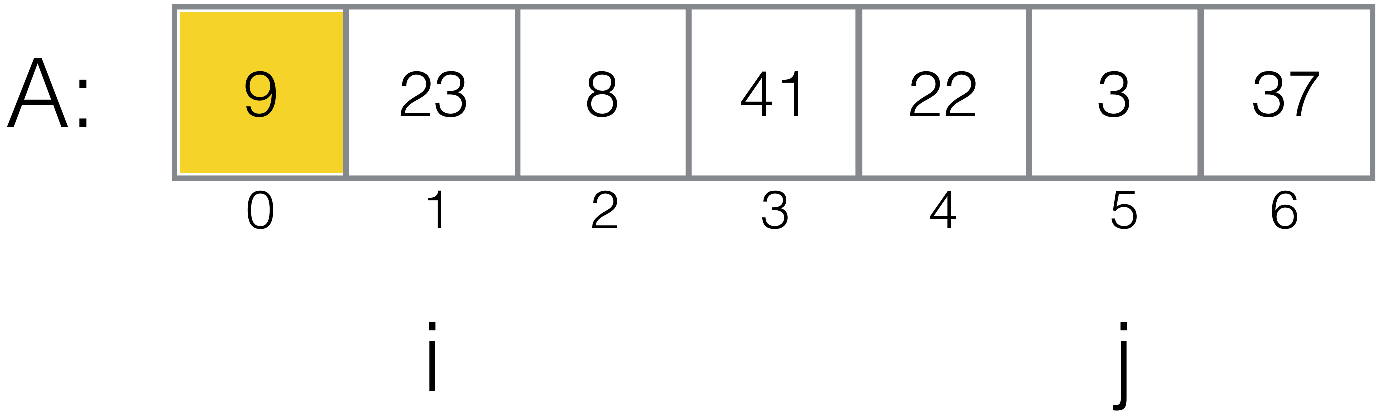
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Hoare Partitioning

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



A:	9	23	8	41	22	3	37
	0	1	2	3	4	5	6
		i				j	

p: 9

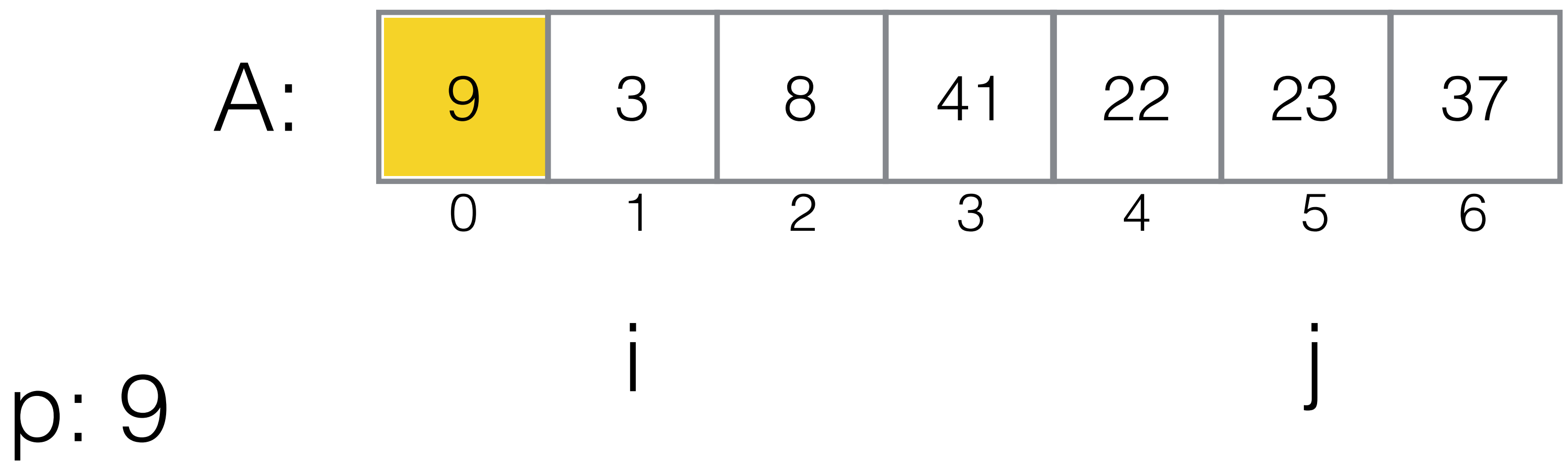
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Hoare Partitioning

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



A:	9	3	8	41	22	23	37
	0	1	2	3	4	5	6
		i				j	

p: 9

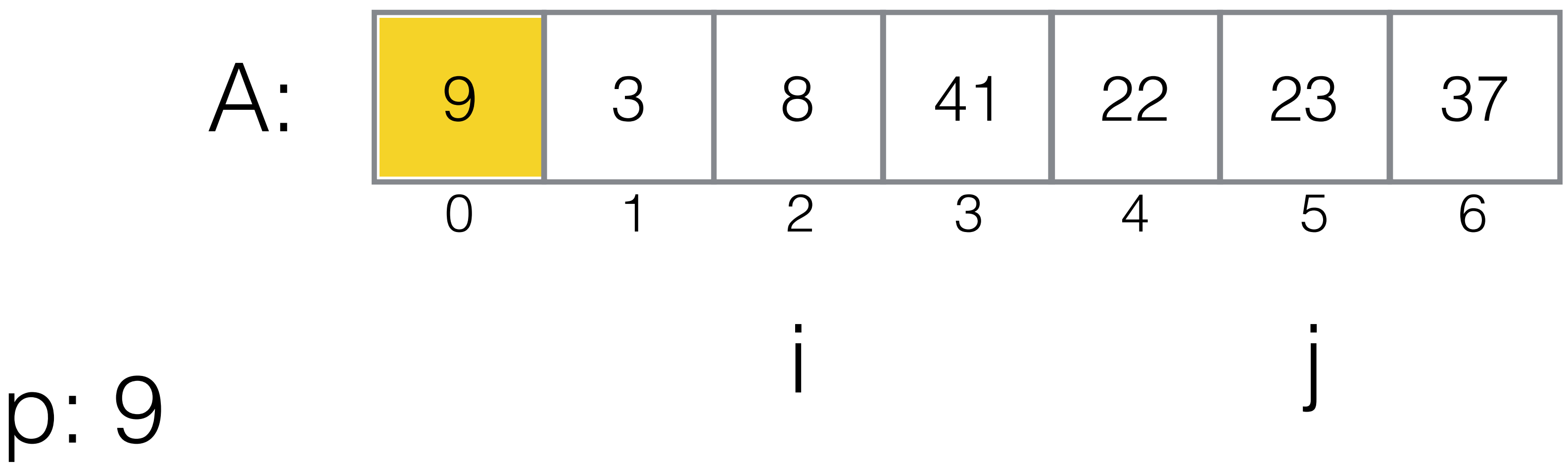
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Hoare Partitioning

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



A:	9	3	8	41	22	23	37
	0	1	2	3	4	5	6
			i			j	

p: 9

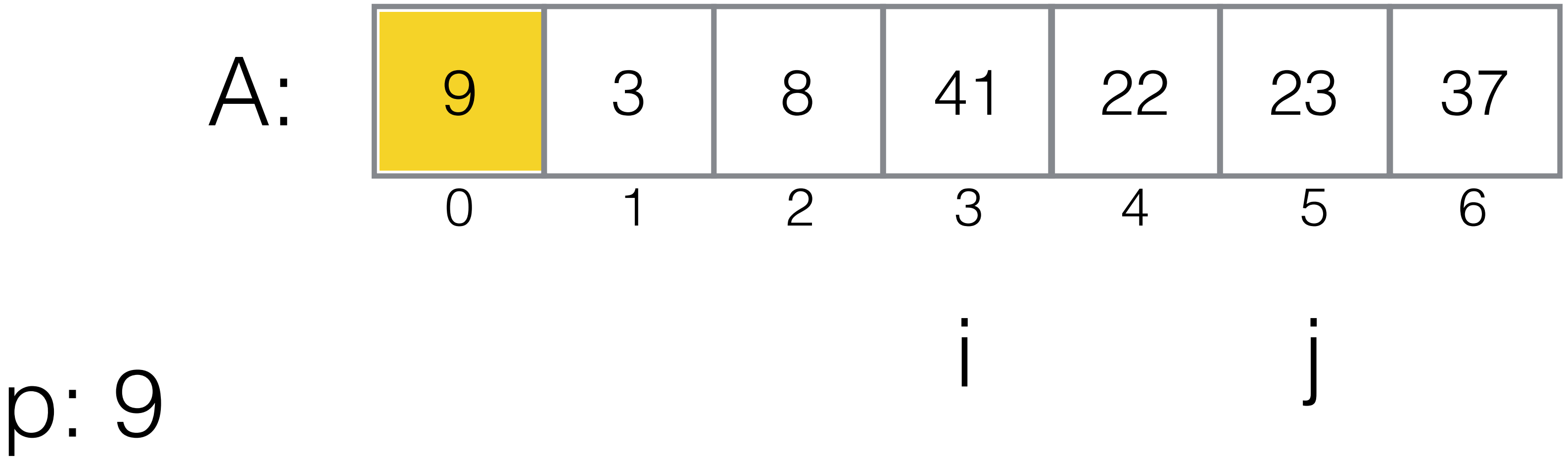
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Hoare Partitioning

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



A:

9	3	8	41	22	23	37
0	1	2	3	4	5	6
			i		j	

p: 9

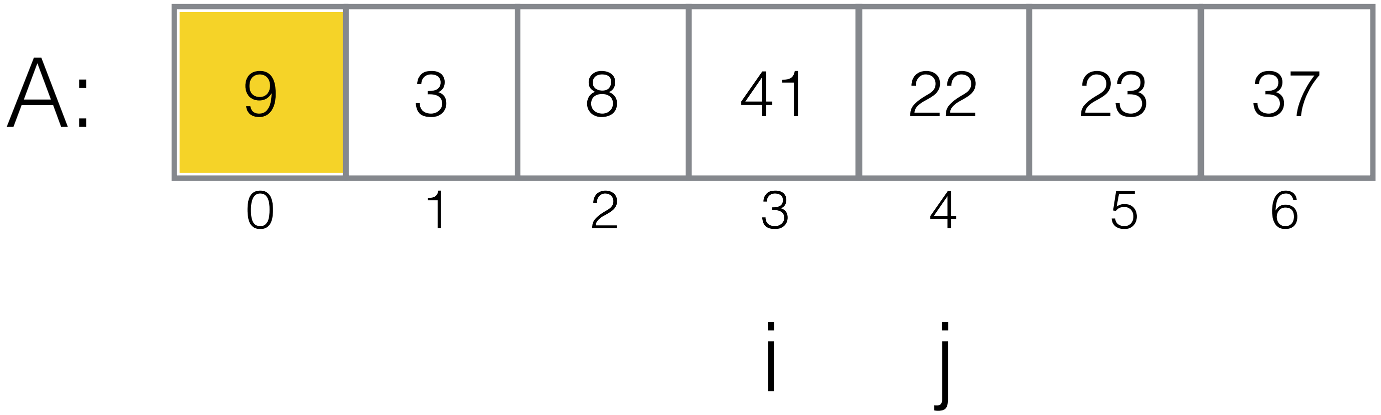
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Hoare Partitioning

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



A:	9	3	8	41	22	23	37
	0	1	2	3	4	5	6
				i	j		

p: 9

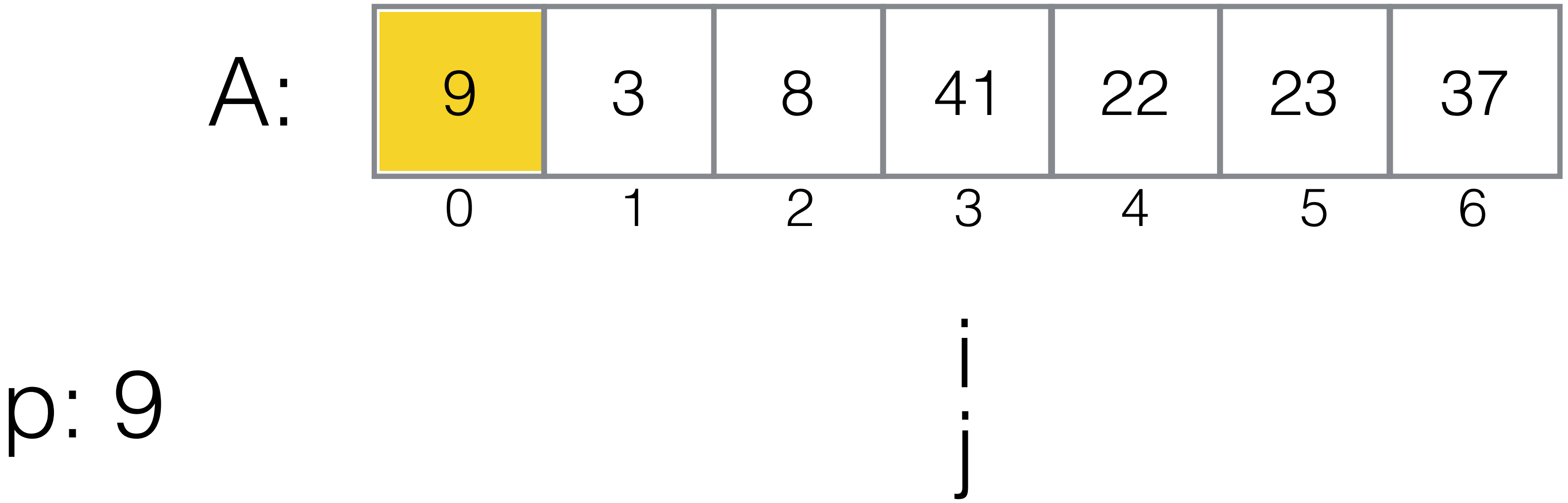
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Hoare Partitioning

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



A:	9	3	8	41	22	23	37
	0	1	2	3	4	5	6

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Hoare Partitioning

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

A:	9	3	8	41	22	23	37
	0	1	2	3	4	5	6
			j	i			

p: 9

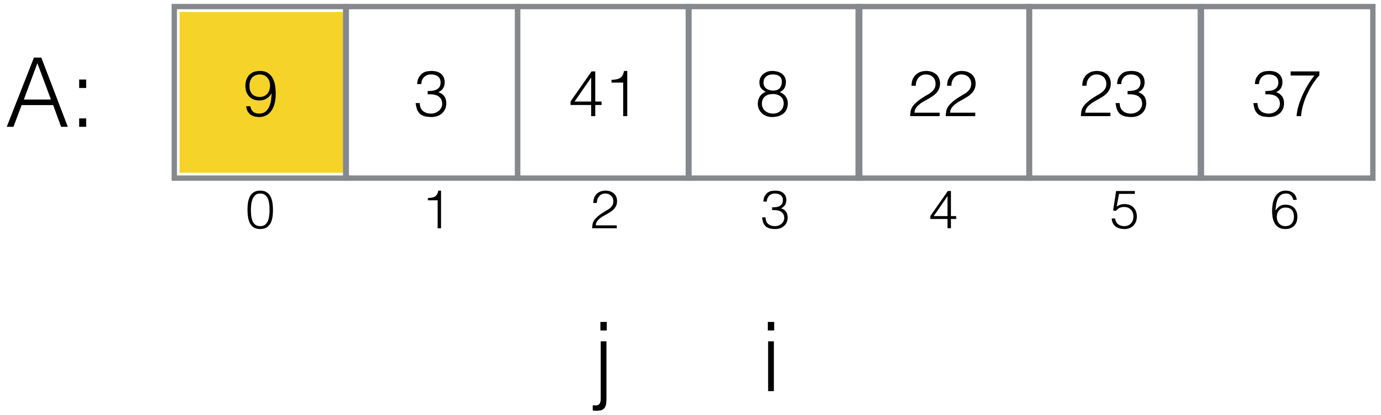
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Hoare Partitioning

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



A:	9	3	41	8	22	23	37
	0	1	2	3	4	5	6
			j	i			

p: 9

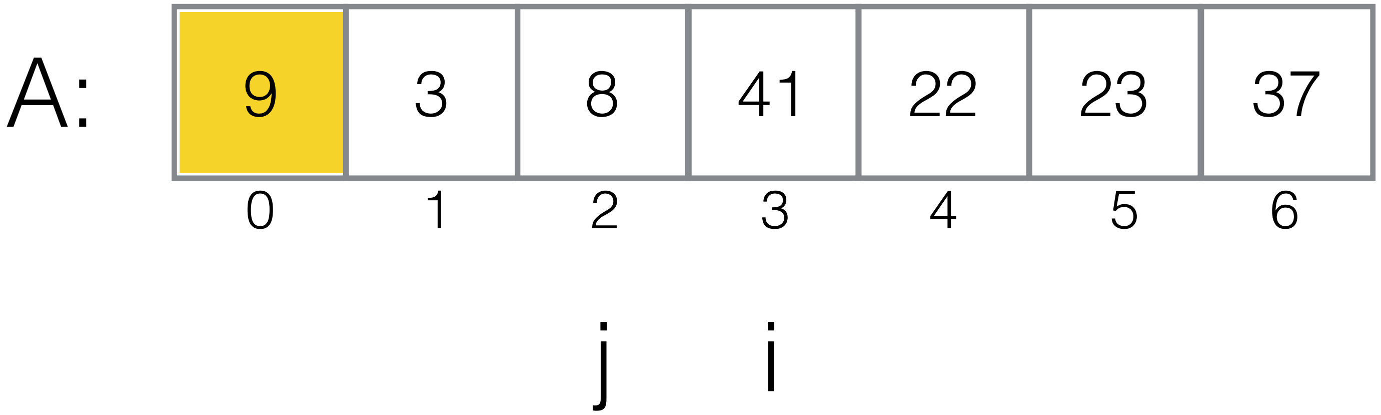
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Hoare Partitioning

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



A:	9	3	8	41	22	23	37
	0	1	2	3	4	5	6
			j	i			

p: 9

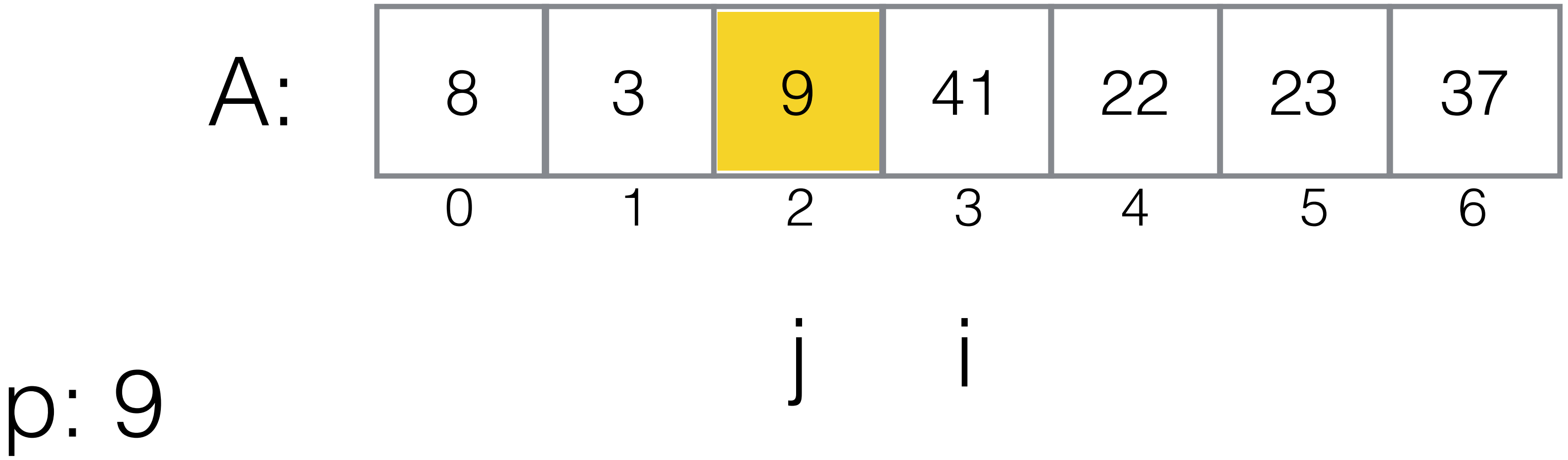
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Hoare Partitioning

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



A:	8	3	9	41	22	23	37
	0	1	2	3	4	5	6
			j	i			

p: 9

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort Analysis: Best Case Analysis

- The best case happens when the pivot is the median; that results in two sub-tasks of equal size.

Assignment Project Exam Help

The ‘n’ is for the https://eduassistpro.github.io/ ns performed by Partition. Add WeChat edu_assist_pro

- By the Master Theorem, $C_{best}(n) \in \Theta(n \log n)$, just as for mergesort, so quicksort’s best case is (asymptotically) no better than mergesort’s worst case.

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort Worst Case

A:

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort Analysis: Worst Case Analysis

- The worst case happens if the array is already sorted.

- In that case, we don't really have divide-and-conquer, becaus e call deals with a problem size that decremented by 1:

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

- That is, $C_{worst}(n) = n + (n - 1) + \cdot \cdot \cdot + 3 + 2 \in \Theta(n^2)$.

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort Improvements: Median-of-Three

- It would be better if the pivot was chosen randomly.

- A cheap and useful approximation to this is to take the median of three candidates, A[lo], A[hi], and A[$\lfloor(lo + hi)/2\rfloor$].

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

- Reorganise the three elements so that p_1 is the median, and p_3 is the largest of the three.

- Now run quicksort as before.

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort Improvements: Median-of-Three

- In fact, with median-of-three, we can have a much faster version than before, simplifying tests in the innermost loops:

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort Improvements: Early Cut-Off

- A second useful improvement is to stop quicksort early and switch to insertion sort. This is easily implemented:

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort Properties

- With these (and other) improvements, quicksort is considered the best available sorting method for arrays of random data.

- A major reason for its speed is the very tight inner loop in PARTITION.

- Although mergesort has a better performance guarantee, quicksort is faster on average.

- In the best case, we get ⌈log2 n⌉ re vels. It can be shown that on random data, the expected number is 2 $\log_e n \approx 1.38 \log_2 n$. So quicksort’s average behaviour is very close to the best-case behaviour.

- Is quicksort stable? ?

- Is it in-place? yes

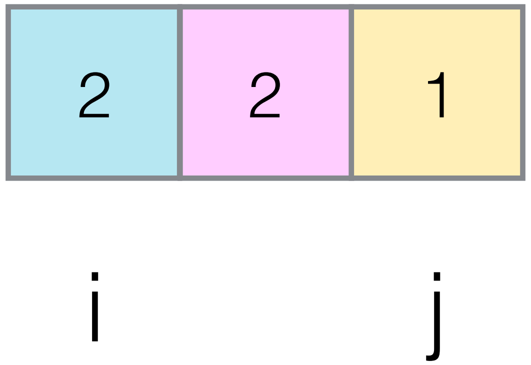
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort Stability



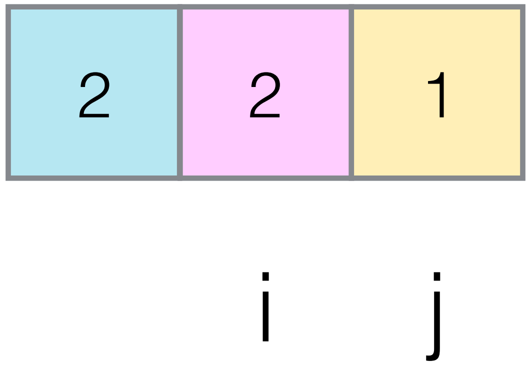
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort Stability



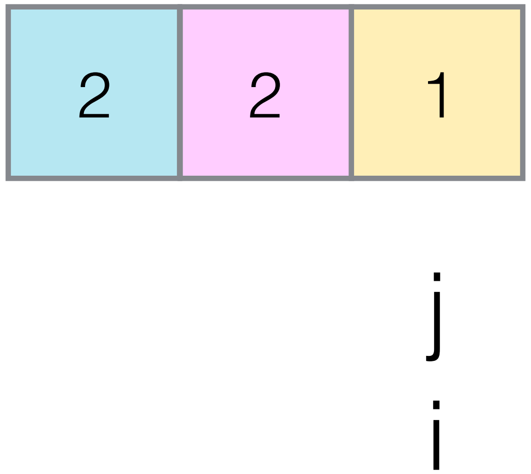
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort Stability



2 2 1

j

i

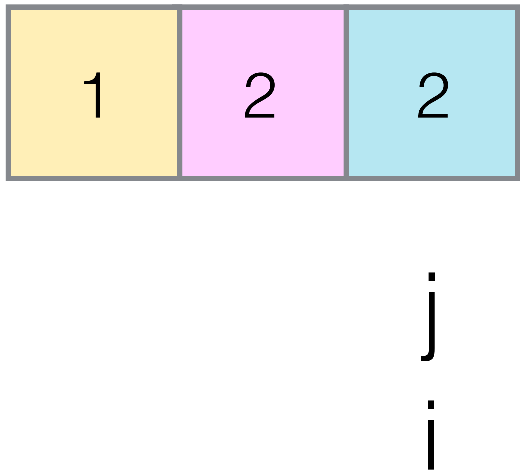
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort Stability



Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort Stability

1	2	2

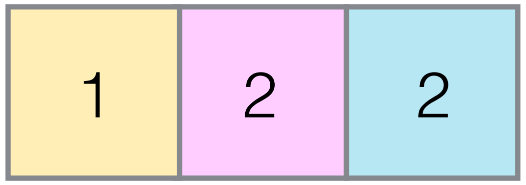
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

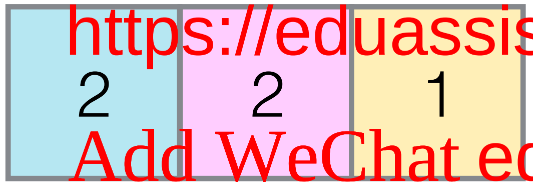
Quicksort Stability



This is where we finished

Assignment Project Exam Help

https://eduassistpro.github.io/



Add WeChat edu_assist_pro

This is where we started

Not stable

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Quicksort Properties

- With these (and other) improvements, quicksort is considered the best available sorting method for arrays of random data.

- A major reason for its speed is the very tight inner loop in PARTITION.

- Although mergesort has a better performance guarantee, quicksort is faster on average.

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

- In the best case, we get $\lceil \log_2 n \rceil$ recursion levels. It can be shown that on random data, the expected number is $2 \log_e n \approx 1.38 \log_2 n$. So quicksort's average behaviour is very close to the best-case behaviour.

- Is quicksort stable? no

- Is it in-place? yes

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Next up

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

- Tree traversal m apply the divide-and-conquer technique to est-pair problem.

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License