

# Information Science Model Answers of Sorting and String

Naonori Kakimura

垣村尚徳

kakimura@global.c.u-tokyo.ac.jp

2017/01/07



# Contents

---

- Sorting exercises
  - Behavior
  - Making a graph
- Dealing with String: Match and reverse

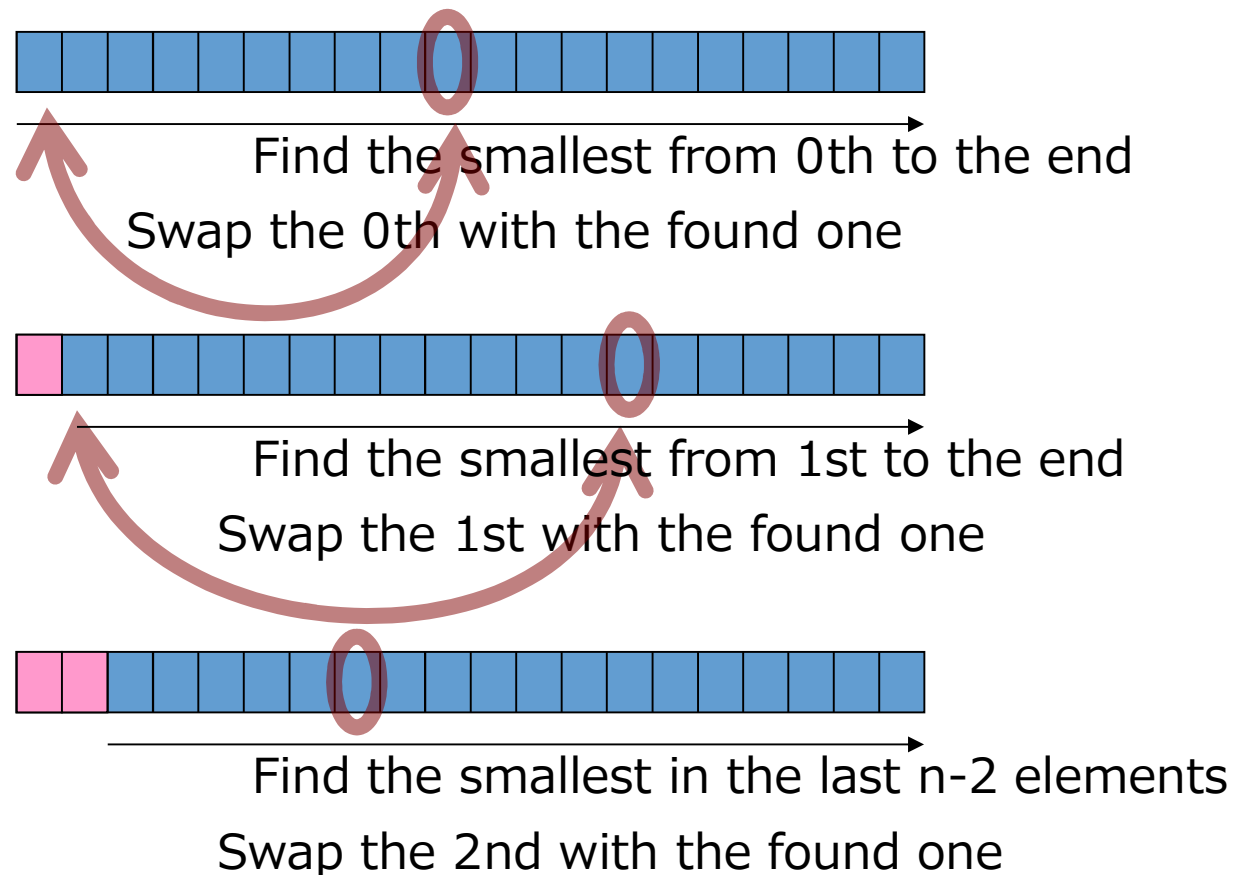
# Review of Sorting Exercises

3

## ➤ Simple Sort Algorithm

- Repeat the following

- Find the minimum index of the remaining elements
- Swap it with the head and delete it



# Simple Sort

---

4

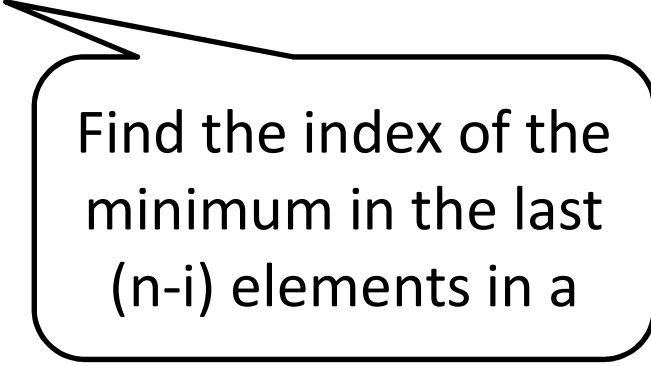
```
def simplesort(a)
  for i in 0..(a.length()-1)
    k = min_index(a,i)
    v = a[k]
    a[k] = a[i]
    a[i] = v
  end
  a
end
```

# Simple Sort

---

5

```
def simplesort(a)
  for i in 0..(a.length()-1)
    k = min_index(a,i)
    v = a[k]
    a[k] = a[i]
    a[i] = v
  end
  a
end
```



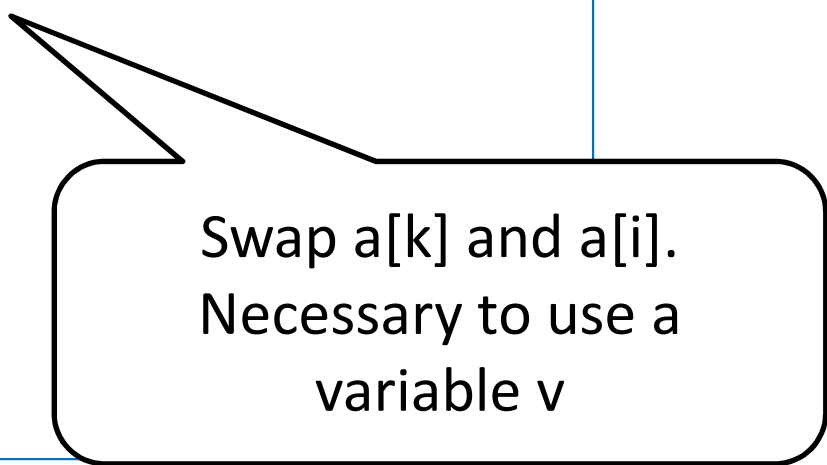
Find the index of the minimum in the last (n-i) elements in a

# Simple Sort

---

6

```
def simplesort(a)
  for i in 0..(a.length()-1)
    k = min_index(a,i)
    v = a[k]
    a[k] = a[i]
    a[i] = v
  end
  a
end
```



Swap a[k] and a[i].  
Necessary to use a  
variable v

# Exercise: Apply to $a=[1,4,2,9,8,3,2,6,4]$

7

It repeats 9 times(not 6)

Do nothing at  $i=0$  (1 is at the beginning)

Swap min and head  
(don't change the others)

8 in this case

```
def simplesort(a)
  for i in 0..(a.length()-1)
    # display a at this point
    k = min_index(a,i)
    v = a[k]
    a[k] = a[i]
    a[i] = v
  end
  a
end
```

i	
0	$a=[1,4,2,9,8,3,2,6,4]$
1	$a=[1,4,2,9,8,3,2,6,4]$
2	$a=[1,2,4,9,8,3,2,6,4]$
3	$a=[1,2,9,8,3,4,6,4]$
4	$a=[1,2,2,3,8,9,4,6,4]$
5	$a=[1,2,2,3,4,9,8,6,4]$
6	$a=[1,2,2,3,4,4,8,6,9]$
7	$a=[1,2,2,3,4,4,6,8,9]$
8	$a=[1,2,2,3,4,4,6,8,9]$
output	$a=[1,2,2,3,4,4,6,8,9]$

# min\_index(a, i)

8

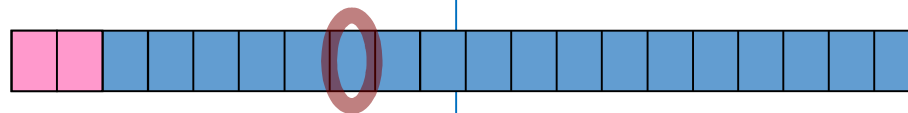
- Find the index with minimum value among  $a[i]..a[a.length()-1]$

```
def min_index(a, i)
  m = i
  for p in i..(a.length()-1)
    if a[p] < a[m]
      m = p
    end
  end
  m
end
```

m: index w/ minimum value so far

check all entries in  $p=i..$ the last

If p-th one is smaller, set m to be p.



Find the smallest in the last  $n-i+1$  elements

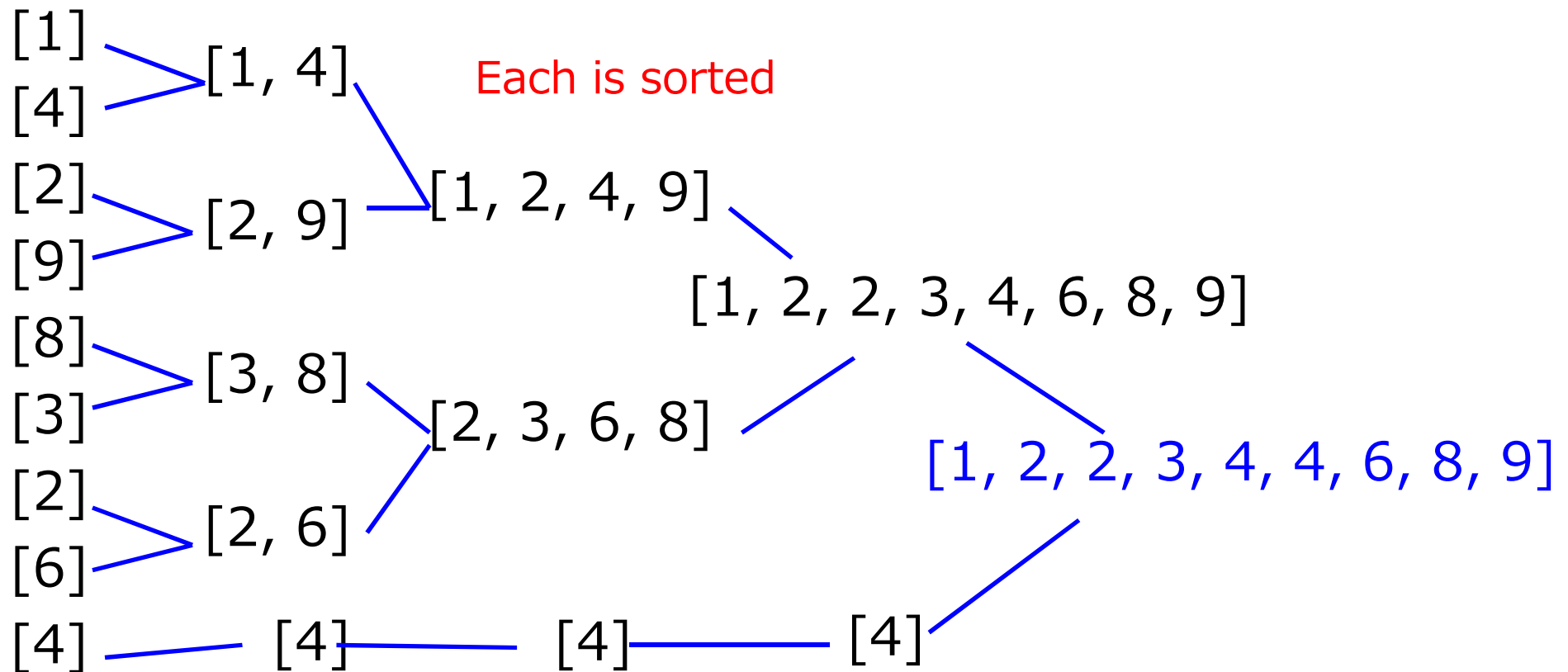
Swap the i-th with the found one



# Exercise: Apply merge Sort

9

➤  $a = [1, 4, 2, 9, 8, 3, 2, 6, 4]$



Begin with 9 arrays

# Exercise: Apply merge Sort

10

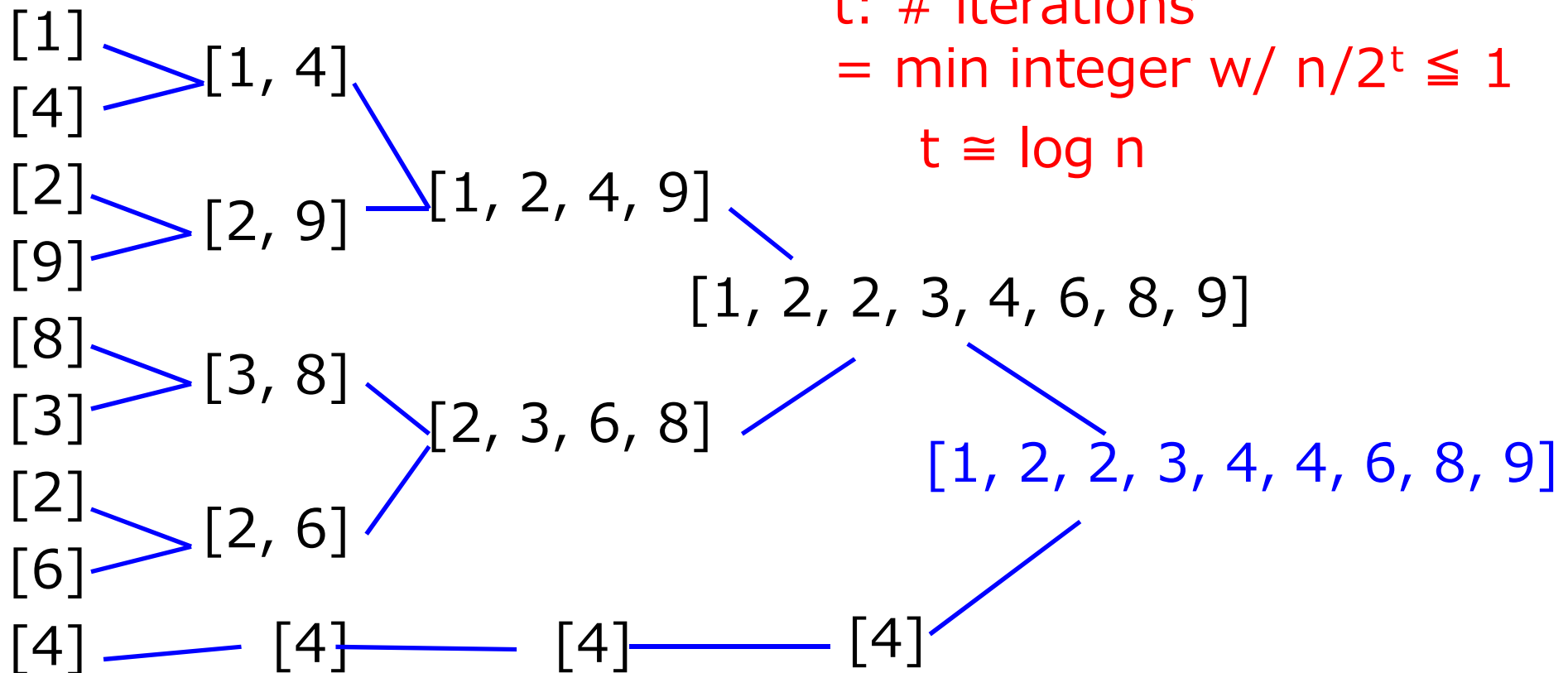
- #arrays reduces by almost half in each iteration
  - $9(\cong n) \rightarrow 5(\cong n/2) \rightarrow 3(\cong n/4) \rightarrow 2(\cong n/8) \rightarrow 1(\cong n/16)$

Min integer at least  $n/2$

t: # iterations

= min integer w/  $n/2^t \leq 1$

$t \cong \log n$



- Sorting exercises
  - Behavior
  - Making a graph
  
- Dealing with String: Match and reverse

# Remarks on Exercise 4

---

12

- Make a graph of the computational times for simple sort and merge sort

## ➤ Remark

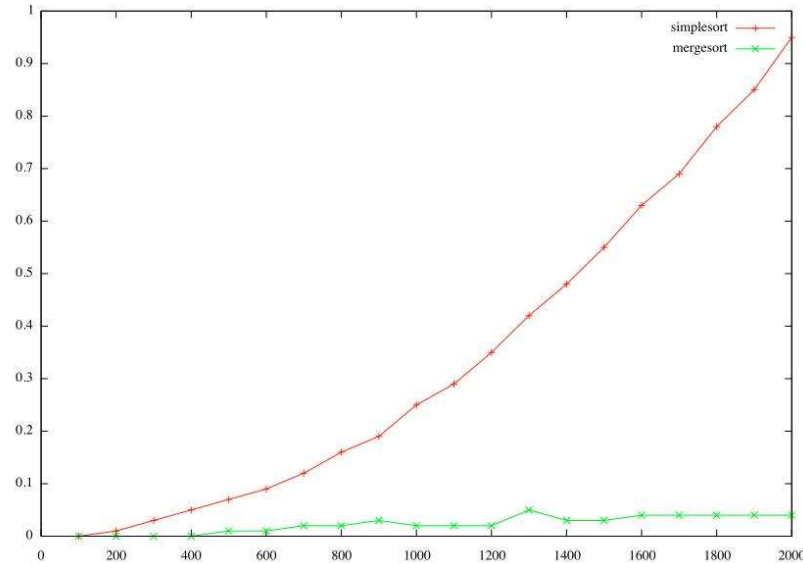
- Function “bench” does not work at home
  - have to install new software “gnuplot”

# How to Read Log Scale Graph

13

## ► Time

look like



$$O(n^2)$$

$$O(n \log n)$$

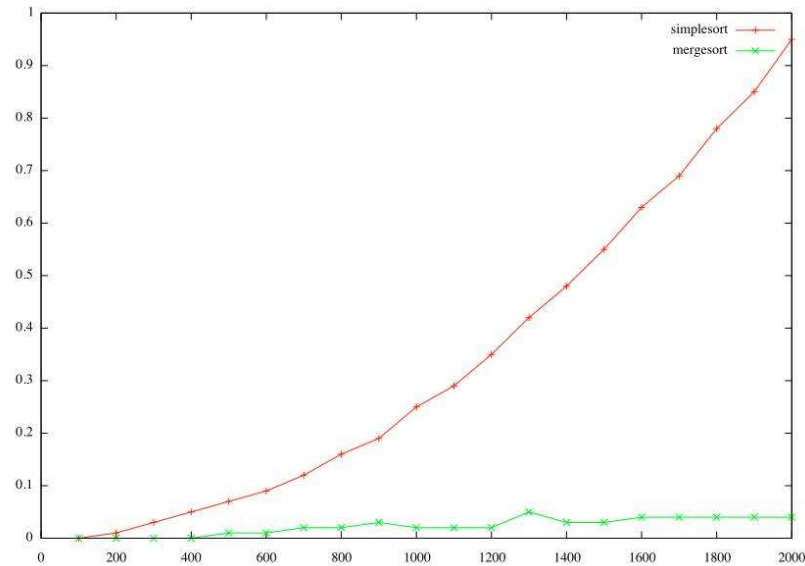
## ► How can we analyze more precisely?

● → log-scale graph

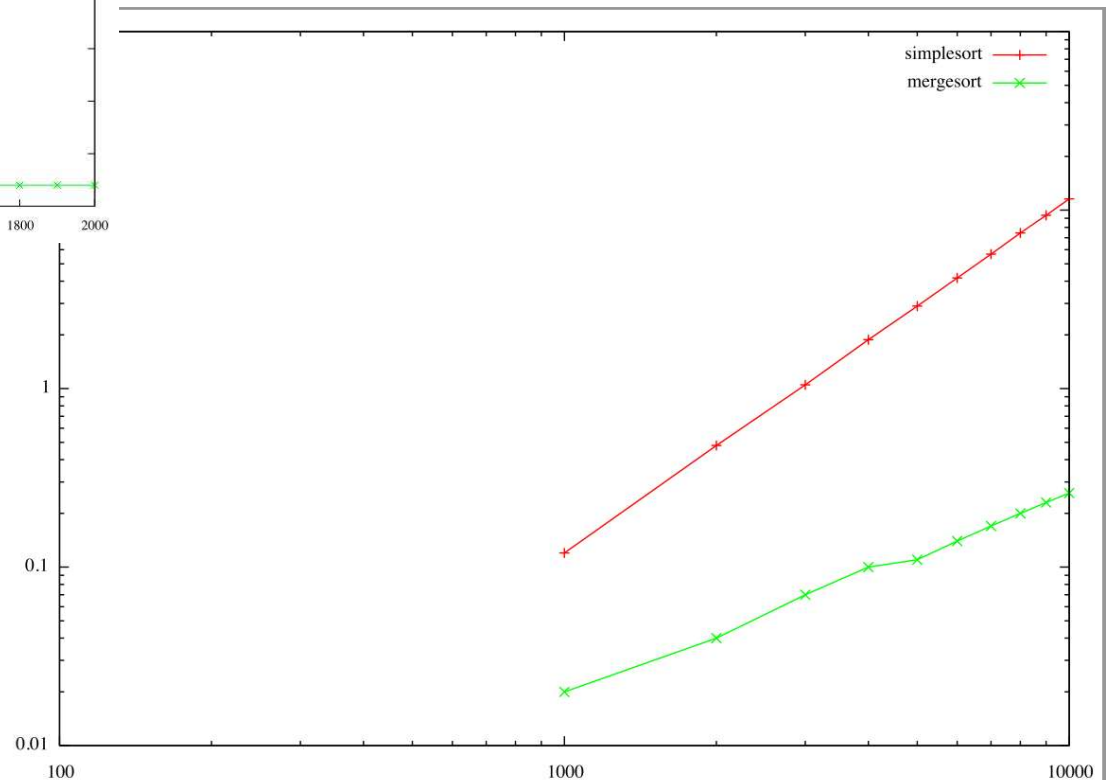
# How to Read Log Scale Graph

14

## ► Time



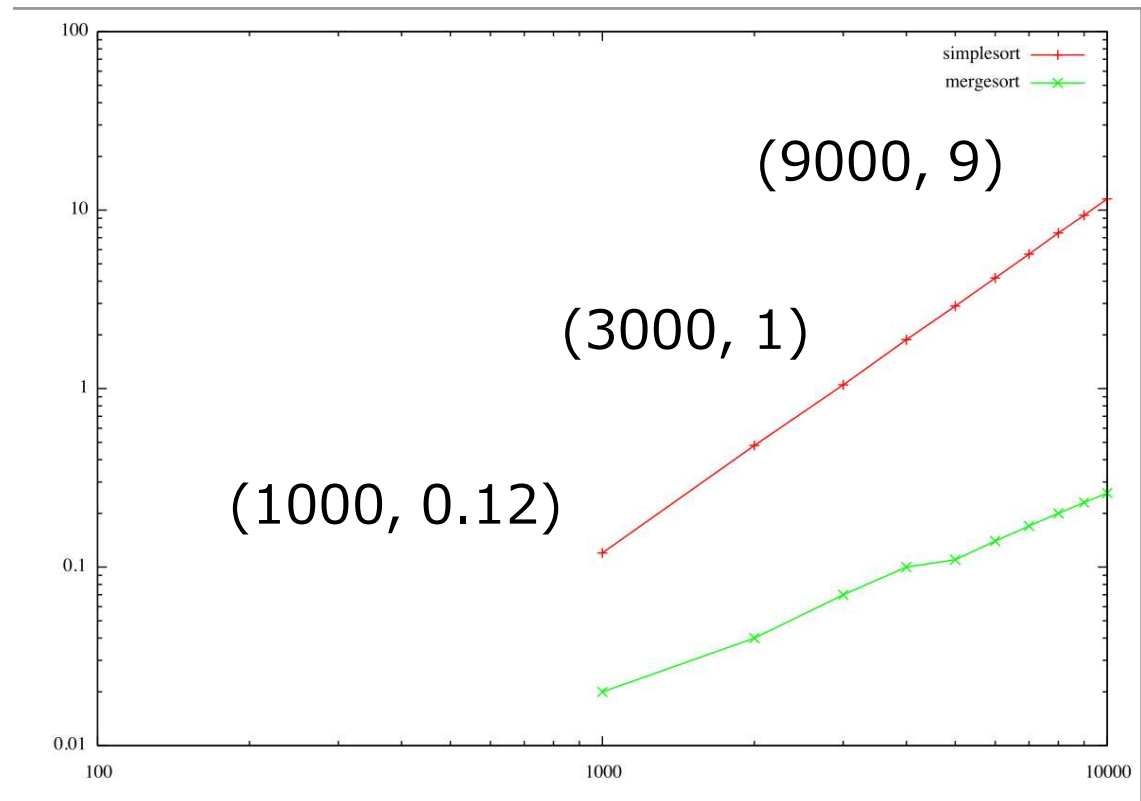
Log-scale graph



# Simple Sort Graph

15

- Compute the slope using representatives



# Simple Sort Graph

16

## ➤ Slope

$$\frac{\log 1 - \log 0.12}{\log 3000 - \log 1000} = 1.92..$$

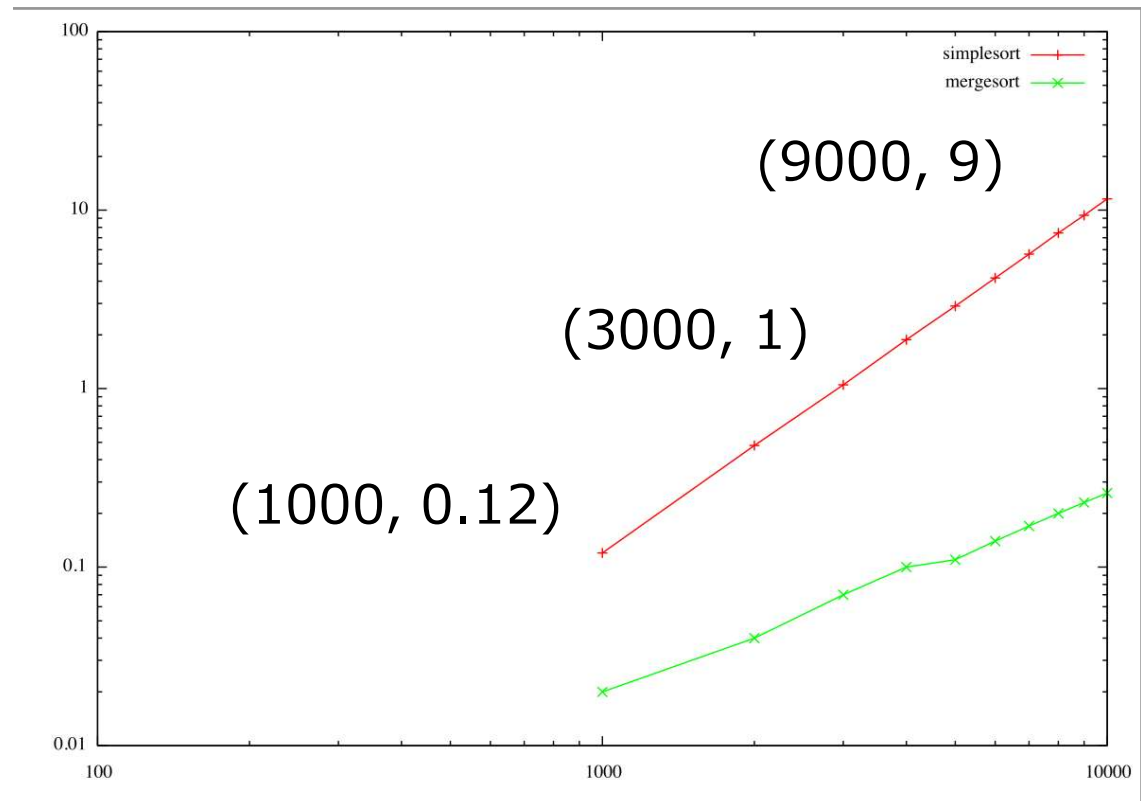
$$\frac{\log 9 - \log 1}{\log 9000 - \log 3000} = 2$$

## ➤ Implying that

$$\log(\text{time}) \propto 2 \log n$$

$$\text{time} \propto n^2$$

$$O(n^2)$$





# Merge-Sort Graph

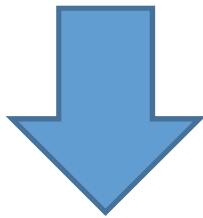
17

- Unlikely to a straight line

$$\frac{\log 0.07 - \log 0.03}{\log 4000 - \log 3000} = 2.94..$$

$$\frac{\log 0.11 - \log 0.03}{\log 8000 - \log 3000} = 1.32..$$

Slope is  $> 1$ , but  $< 2$  when the size is large

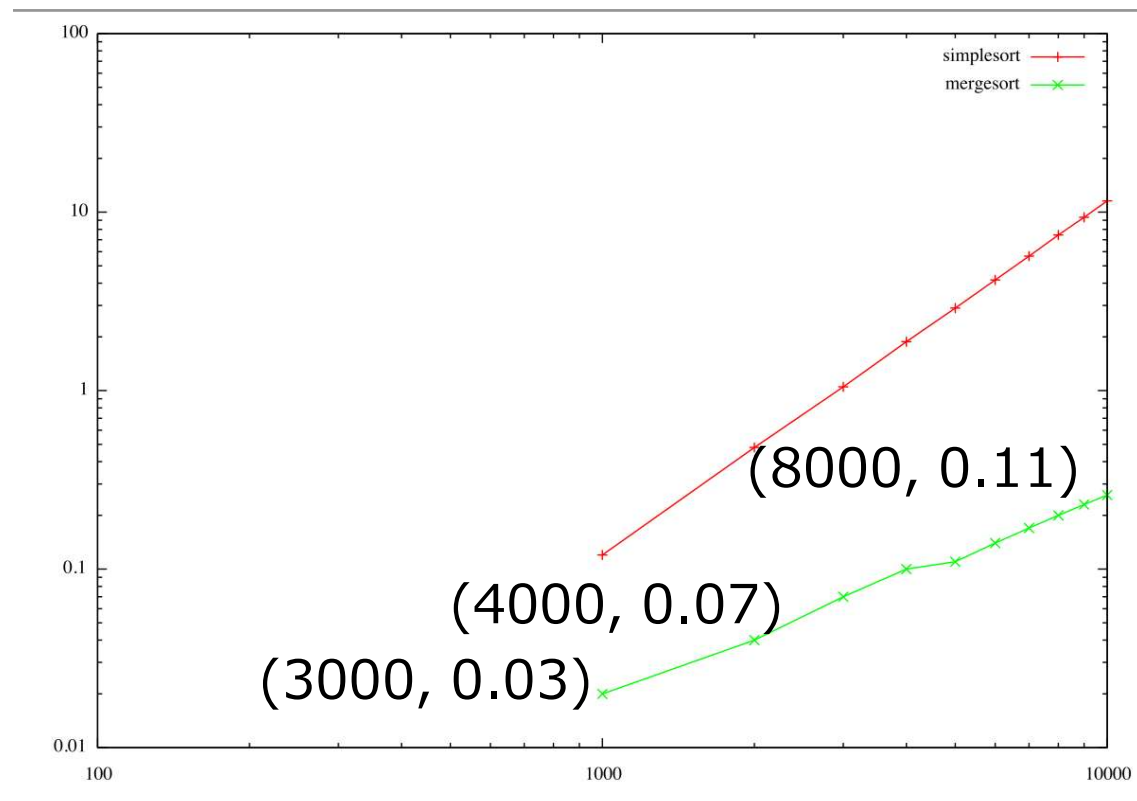


worse than  $O(n)$   
faster than  $O(n^2)$

Close to theory

time  $\propto n \log n$

Might be better  
to plot "time"/ $n$



- Sorting exercises
  - Behavior
  - Making a graph
- Dealing with String: Match and reverse

# Last Exercise1:Searching a Substring

---

19

- Find some keyword in a string
  - Ex. Analysis of DNA sequences

	0	1	2	3	4	5	6	7	8
s:	b	a	l	a	l	a	i	k	a

Q. Is there “alai”, denoted by  $p$ , in the sequence?

# (review) Searching Keywords

20

```
def match(s,p)
  i = 0
  w = p.length()
  while submatch(s,i,p,w) < w
    i = i + 1
  end
  i
end
```

Function that counts how many characters are matched from the i-th char in s

Decide if  
# matched = length of p

Downloadable from the Ruby-program website

# (review) Procedure submatch(s,i,p,w)

21

```
def submatch (s,i,p,w)
```

```
  j = 0
```

```
  while j < w && s[(i+j)..(i+j)] == p[j..j]
```

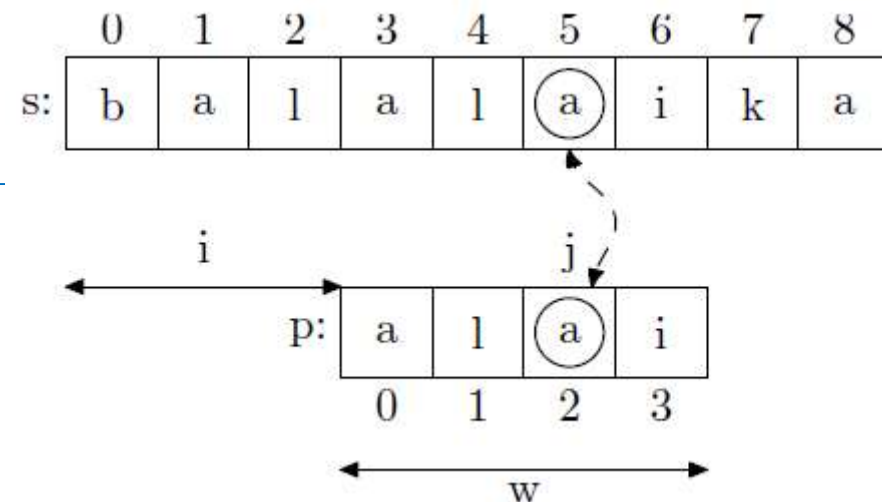
```
    j = j + 1
```

```
end
```

```
  j
```

```
end
```

Compare  
the (i+j)th in s and the jth in p



while they coincide  
we increment j by one

# Last Exercise 1:

---

22

- What is the computational complexity of the function match?
  - Let  $n$  be the length of  $s$ , and  $m$  be the length of  $p$ ,
  - Represent the complexity using  $n$  and  $m$
  
- **Hint:** It suffices to consider the worst case:
  - Consider the following case & estimate #repetitions
    - $s = [aa \cdots aaaab]$ 

$\underbrace{\hspace{1.5cm}}$   
 $n-1$
    - $p = [aa \cdots ab]$ 

$\underbrace{\hspace{1.5cm}}$   
 $m-1$

# Computational Time of Match

23

- Repeatedly compare  $s[i..(i+w-1)]$  and  $p$

s: 

a	a	a	a	a	a	a	a	a
---	---	---	---	---	---	---	---	---

 a a a a a a a a a a a b

p: a a a a b

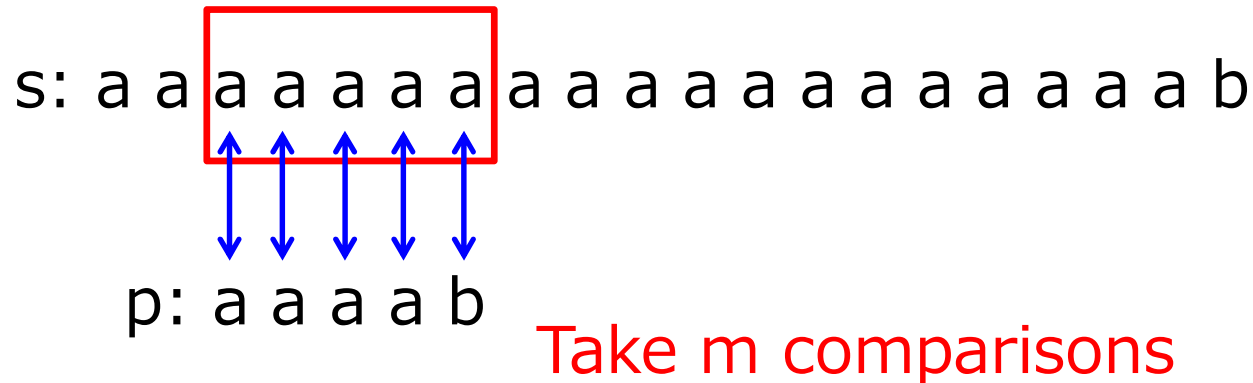
- When  $s=[aa\cdots aaaab]$  and  $p=[aa\cdots ab]$ 
  - Need to consider  $s[i..(i+m-1)]$  for all  $i=0,\cdots,n-m$ 
    - $\#(\text{substrings of size } m \text{ in } s) = n - m + 1$
  - Each time we call submatch

Computational time:  
 $(n-m+1)*[\text{time for submatch}]$

# Behavior of submatch (when fixing i)

24

- Compare each element in  $s[i..(i+w-1)]$  and  $p$



- When  $s=[aa\cdots aaaab]$  and  $p=[aa\cdots ab]$ 
  - Need to consider  $s[i..(i+m-1)]$  for all  $i=0,\cdots,n-m$ 
    - # substrings ( $[aaa\cdots a]$  or  $[aaa\cdots b]$ ) =  $n - m + 1$
  - Each time we call submatch

$$\begin{aligned}\text{total time: } & (n-m+1)*m \quad (\leftarrow \text{ok as an answer}) \\ & = O(n*m) \quad (\text{leave only the most dominant term})\end{aligned}$$



# Last Exercise 2: Reverse a String

---

25

- Given a string, we want to reverse it
  - Ex. For `s="abcdef"`, the output is `"fedcba"`
  
- Requirement
  - Use `"while"` (or `"for"`)
  - Do not use `s.reverse()` nor `s.split("").reverse().join()`
    - Already implemented in Ruby

## Last Exercise 2: Framework

---

26

➤ Fill in the question marks

```
def reverse(s)
  result = "" # empty string(length 0)
  i = ??
  while i >= 0 do
    ??
    i = i - 1
  end
  result # return the reversed string
end
```

## ➤ Idea

```
def reverse(s)
  result = ""
  i = s.length()-1
  while i >= 0 do
    ??
    i = i - 1
  end
  result
end
```

Read s one-by-one from the end



s: P E A K 2 0 1 3

# The first entry is the end index

# decrease by one

(final state)

result: 3 1 0 2 K E A P

## ➤ Idea

```
def reverse(s)
  result = ""
  i = s.length()-1
  while i >= 0 do
    ??
    i = i - 1
  end
  result
end
```

Read s one-by-one from the end



s: P E A K 2 0 1 3

result: 3 1 0...  
(start from empty)

(final state)

result: 3 1 0 2 K E A P

## ➤ Idea

```
def reverse(s)
  result = ""
  i = s.length()-1
  while i >= 0 do
    ??
    i = i - 1
  end
  result
end
```

i 7(s.length-1)

s: P E A K 2 0 1 3

result: 3 1

Append s[i..i] at the end

result = "3" + "1"

result  
in the previous iteration

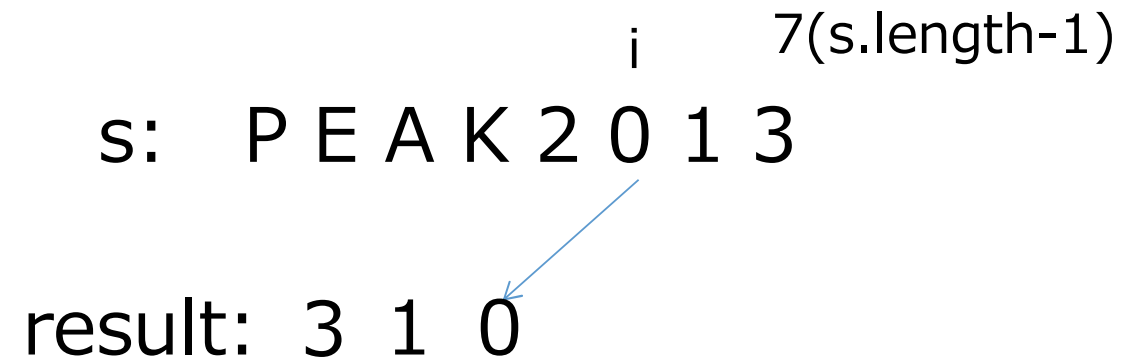
## ➤ Idea

```
def reverse(s)
  result = ""
  i = s.length()-1
  while i >= 0 do
    ??
    i = i - 1
  end
  result
end
```

$i$        $7(s.length-1)$

s: P E A K 2 0 1 3

result: 3 1 0



Append s[i..i] at the end

result = "31" + "0"

result  
in the previous iteration

## Exercise 2: Reverse a String

31

### ➤ Another solution: Using concatenation

```
def reverse(s)
  result = ""
  i = s.length()-1
  while i >= 0 do
    result = result + s[i..i]
    i = i - 1
  end
  result
end
```

OK to be s[i]

s: P E A K 2 0 1 3

i

“result”+“0”  
(=“31”)  
String so far obtained

“result” contains the reversed string so far  
add s[i..i] at the end of “result”

## Exercise 2: Recursive Ones

---

32

### ➤ Based on recursion

```
def reverse_r(s)
  n = s.length()
  if n == 1
    s
  else
    s[n-1..n-1] + reverse_r(s[0..(n-2)])
  end
end
```

Ex.

(Reverse of "PEAK2013") = 3 + (Reverse of "PEAK201")



# Incorrect Program from some of you

---

33

```
def reverse_wrong(s)
  result = ""
  i = s.length()-1
  while i >= 0 do
    c=s[i]
    s[i]=s[s.length-1-i]
    s[s.length-1-i]=c
    i = i - 1
  end
  result
end
```

Prepare result

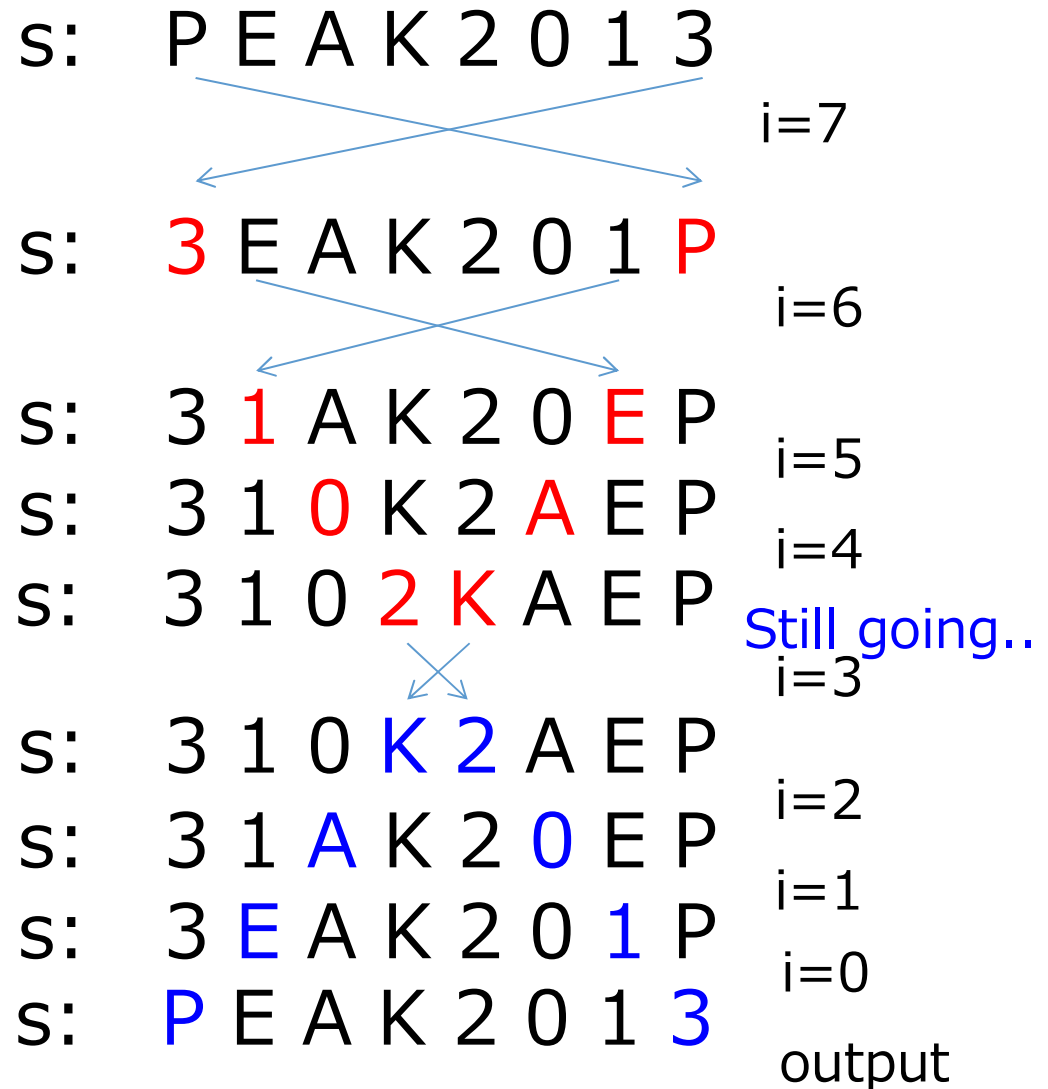
Because we return "result",  
need to change "result" not "s"

# Analysis of Program from some of you

34

$7(s.length-1)$

```
def reverse_wrong(s)
  result = ""
  i = s.length()-1
  while i >= 0 do
    c=s[i]
    s[i]=s[s.length-1-i]
    s[s.length-1-i]=c
    i = i - 1
  end
  result
end
```



# To make it correct: Run to half

35

$7(s.length-1)$

```
def reverse2(s)
  i = s.length() - 1
  k = s.length()/2
  while i >= k do
    # stop when i=k
    c = s[i]
    s[i] = s[s.length-1-i]
    s[s.length-1-i] = c
    i = i - 1
  end
  s # change output
end
```

s: P E A K 2 0 1 3      i=7

s: 3 E A K 2 0 1 P      i=6

s: 3 1 A K 2 0 E P      i=5

s: 3 1 0 K 2 A E P      i=4

s: 3 1 0 2 K A E P      i=3

Stop here (when i=3)