Algorithms

Assignment #2

Yoon Suk Kang
School of Computer Science
Chungbuk National University

Assignment #2: Description

❑ Overview

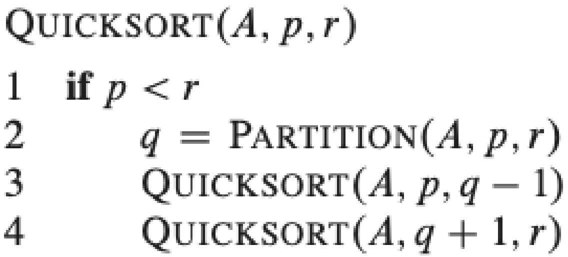
- Sorting a list of unordered numbers **in ascending order** using **sorting algorithms**

❑ 3 sorting algorithms to implement

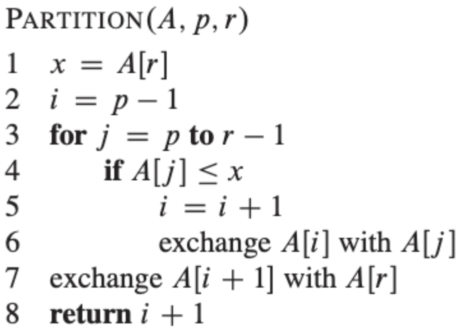
- Naïve quick sort
- Randomized quick sort
- Randomized quick (median-of-3) sort

Assignment #2: Description (Cont.)

❑ Naïve quick sort



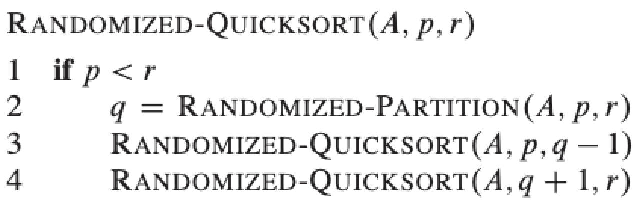
```
QUICKSORT$(A, p, r)$
if $p < r$
    $q = $ PARTITION$(A, p, r)$
    QUICKSORT$(A, p, q - 1)$
    QUICKSORT$(A, q + 1, r)$
```



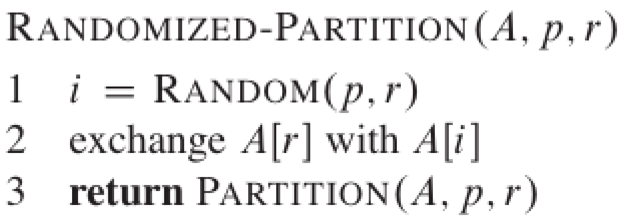
```
PARTITION$(A, p, r)$
$x = A[r]$
$i = p - 1$
for $j = p$ to $r - 1$
    if $A[j] \leq x$
        $i = i + 1$
        exchange $A[i]$ with $A[j]$
exchange $A[i + 1]$ with $A[r]$
return $i + 1$
```

Assignment #2: Description (Cont.)

❑ Randomized quick sort



```
RANDOMIZED-QUICKSORT(A, p, r)
if p < r
    q = RANDOMIZED-PARTITION(A, p, r)
    RANDOMIZED-QUICKSORT(A, p, q - 1)
    RANDOMIZED-QUICKSORT(A, q + 1, r)
```

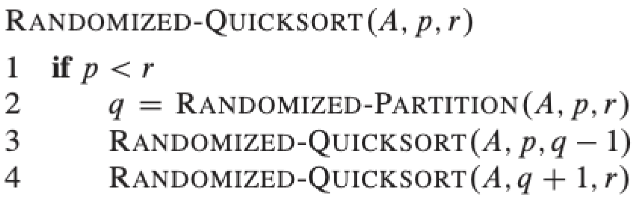


```
RANDOMIZED-PARTITION(A, p, r)
i = RANDOM(p, r)
exchange A[r] with A[i]
return PARTITION(A, p, r)
```

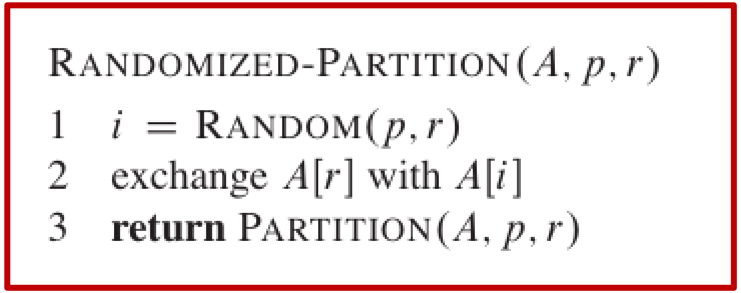
Assignment #2: Description (Cont.)

Randomized quick sort (median-of-3)

- Pick 3 values randomly and use a median value as a pivot



```
RANDOMIZED-QUICKSORT($A, p, r$)
if $p < r$
    $q$ = RANDOMIZED-PARTITION($A, p, r$)
    RANDOMIZED-QUICKSORT($A, p, q - 1$)
    RANDOMIZED-QUICKSORT($A, q + 1, r$)
```

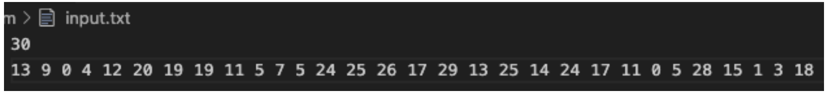


```
RANDOMIZED-PARTITION($A, p, r$)
$i$ = RANDOM($p, r$)
exchange $A[r]$ with $A[i]$
return PARTITION($A, p, r$)
```

Assignment #2: Description (Cont.)

❑ Input file

- The first line has an integer value which represents the length of the unordered list
- The second line has the unordered list that you will sort

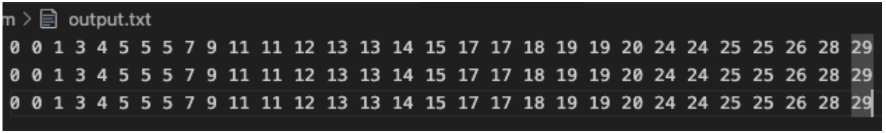


```
30
13 9 0 4 12 20 19 19 11 5 7 5 24 25 26 17 29 13 25 14 24 17 11 0 5 28 15 1 3 18
```

Example of Input.txt

❑ Output file

- The first line is the result of quick sort
- The second line is the result of randomized quick sort
- The last line is the result of randomized quick (median-of-3) sort



```
0 0 1 3 4 5 5 5 7 9 11 11 12 13 13 14 15 17 17 18 19 19 20 24 24 25 25 26 28 29
0 0 1 3 4 5 5 5 7 9 11 11 12 13 13 14 15 17 17 18 19 19 20 24 24 25 25 26 28 29
0 0 1 3 4 5 5 5 7 9 11 11 12 13 13 14 15 17 17 18 19 19 20 24 24 25 25 26 28 29
```

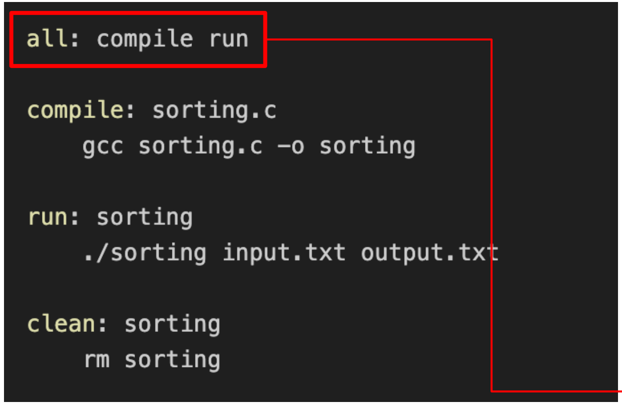
No newline at the end of the output file

No space at the end of each line

Example of output.txt

Assignment #2: Makefile (C/C++)

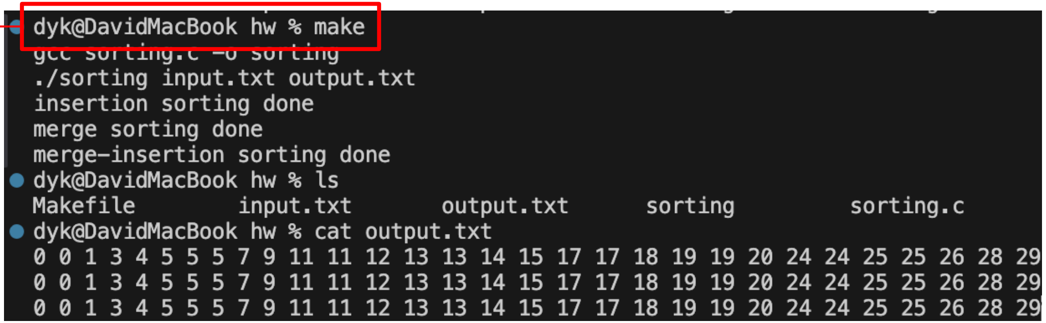
❑ Writing a MakeFile and Test



```
all: compile run

compile: sorting.c
    gcc sorting.c -o sorting

run: sorting
    ./sorting input.txt output.txt

clean: sorting
    rm sorting
```

Makefile

'**make**' command performs 'compile' and 'run'



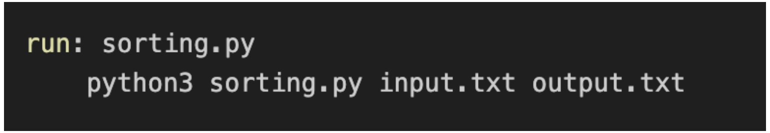
```
dyk@DavidMacBook hw % make
gcc sorting.c -o sorting
./sorting input.txt output.txt
insertion sorting done
merge sorting done
merge-insertion sorting done
dyk@DavidMacBook hw % ls
Makefile        input.txt       output.txt      sorting         sorting.c
dyk@DavidMacBook hw % cat output.txt
0 0 1 3 4 5 5 5 7 9 11 11 12 13 13 14 15 17 17 18 19 19 20 24 24 25 25 26 28 29
0 0 1 3 4 5 5 5 7 9 11 11 12 13 13 14 15 17 17 18 19 19 20 24 24 25 25 26 28 29
0 0 1 3 4 5 5 5 7 9 11 11 12 13 13 14 15 17 17 18 19 19 20 24 24 25 25 26 28 29
```

After executing your code

"output.txt" that includes the results of your code will be generated

Assignment #2: Makefile (Python)

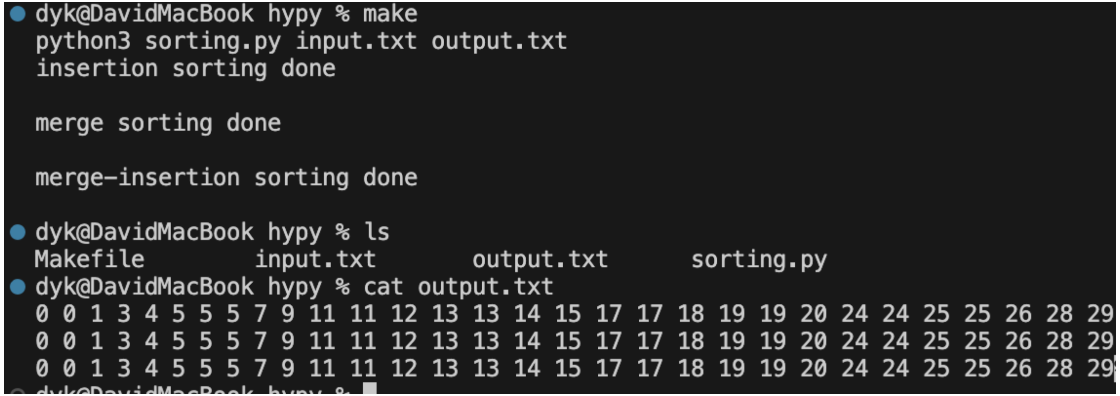
❑ Writing a MakeFile and Test



```
run: sorting.py
    python3 sorting.py input.txt output.txt
```

Makefile

'make’ command performs ‘run’ (python3 sorting.py input.txt output.txt)



```
● dyk@DavidMacBook hypy % make
  python3 sorting.py input.txt output.txt
  insertion sorting done

  merge sorting done

  merge–insertion sorting done

● dyk@DavidMacBook hypy % ls
  Makefile        input.txt       output.txt      sorting.py
● dyk@DavidMacBook hypy % cat output.txt
  0 0 1 3 4 5 5 5 7 9 11 11 12 13 13 14 15 17 17 18 19 19 20 24 24 25 25 26 28 29
  0 0 1 3 4 5 5 5 7 9 11 11 12 13 13 14 15 17 17 18 19 19 20 24 24 25 25 26 28 29
  0 0 1 3 4 5 5 5 7 9 11 11 12 13 13 14 15 17 17 18 19 19 20 24 24 25 25 26 28 29
○ dyk@DavidMacBook hypy % █
```

After executing your code

“output.txt” that includes the results of your code will be generated

Assignment #2: Submission Guideline

❑ Submission

- What: **Compressed file (Source code, makefile)**
  - ❑ Compressed filename: Assignment1_studentID
- Where: **Assignment board on eCampus**
- Deadline: **11:59 PM, October 10th (Thursday), 2024**
  - ❑ +1 day (75%), +2 days (50%), +3 days (**Not accepted**)

❑ Cautions

- Make sure it compiles and runs properly!!
  - ❑ **The size of evaluation input file will vary,** so make sure your code can handle it
  - ❑ If it fails to compile or run during evaluation, it will be considered **as non-functional**
- Comments for all functions/algorithms that you implemented

Q&A:
Assignment #2