Due Date: March 22 (F), BEFORE the class begins

This assignment covers textbook 7.1, 7.2 and Chapter1~4.

1. **QuickSort Algorithm (10 points)**
Exercise 7.1-1 (p173)

2. QuickSort Algorithm (20 points)
Using the PARTITION and QUICKSORT routines in textbook 171, what value of q does each PARTITION return, when all elements in A[1..n] are distinct and sorted in *descending* order? Justify your answer.

3. **QuickSort Algorithm Running Time (20 points)**
Provide tight upper and lower bounds on the running time of the QUICKSORT algorithm (p171) for the above case in Problem 2? Show your answer (the running time) in recurrence and solve the recurrence. Justify your answer.

4. **QuickSort and Substitution (20 points)**
Use the substitution method to prove your answer in 3.

5. **QuickSort Analysis (15 points)**
Assume the partitioning algorithm always produces an 80-to-20 proportional split, write the recurrence of the running time of QuickSort in this case. Solve the recurrence by using a recursion tree.

6. **QuickSort Analysis (15 points)**
Exercise 7.2-5 (p178)

Algorithms -- COMP.4040 Honor Statement
(Courtesy of Prof. Tom Costello and Karen Daniels with modifications)

Must be attached to each submission

Academic achievement is ordinarily evaluated on the basis of work that a student produces independently. Infringement of this Code of Honor entails penalties ranging from reprimand to suspension, dismissal or expulsion from the University.

Your name on any exercise is regarded as assurance and certification that what you are submitting for that exercise is the result of your own thoughts and study. Where collaboration is authorized, you should state very clearly which parts of any assignment were performed with collaboration and name your collaborators.

In writing examinations and quizzes, you are expected and required to respond entirely on the basis of your own memory and capacity, without any assistance whatsoever except such as what is specifically authorized by the instructor.

I certify that the work submitted with this assignment is mine and was generated in a manner consistent with this document, the course academic policy on the course website on Blackboard, and the UMass Lowell academic code.

Date: 03/21/2019

Name (please print): DANGNHI NGO

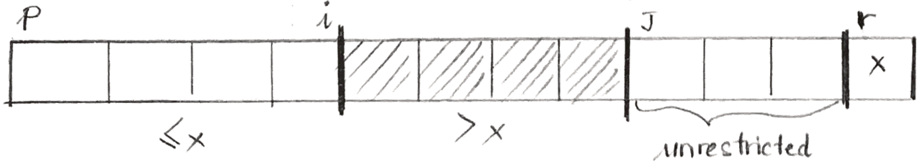
Signature:

١٠/١٠

1/ 10 Quick Sort Algorithm (Exercise 7.1-1)

Illustrate the operation of PARTITION on the array A = {13, 19, 9, 5, 12, 8, 7, 4, 21, 2, 6, 11

	i P,J											r
a/	13	19	9	5	12	8	7	4	21	2	6	11
	i P	J										r
b/	13	19	9	5	12	8	7	4	21	2	6	11
	i P		J									r
c/	13	19	9	5	12	8	7	4	21	2	6	11
	i,P			J								r
d/	9	19	13	5	12	8	7	4	21	2	6	11
	P	i			J							r
e/	9	5	13	19	12	8	7	4	21	2	6	11
	P	i				J						r
f/	9	5	13	19	12	8	7	4	21	2	6	11
	P		i				J					r
g/	9	5	8	19	12	13	7	4	21	2	6	11
	P			i				J				r
h/	9	5	8	7	12	13	19	4	21	2	6	11
	P				i				J			r
i/	9	5	8	7	4	13	19	12	21	2	6	11
	P				i					J		r
J/	9	5	8	7	4	13	19	12	21	2	6	11
	P					i					J	r
k/	9	5	8	7	4	2	19	12	21	13	6	11
	P						i					r
l/	9	5	8	7	4	2	6	12	21	13	19	11
	P						i					r
m/	9	5	8	7	4	2	6	11	21	13	19	12



2/ Quick Sort Algorithm

20 What value of q does each PARTITION return, when all elements in A[1...n] are distinct and sorted in descending order?

Suppose we have an array sorted in descending order: A <5, 4, 3, 2, 1>

q = r is always returned with r is the last index of each subarray

QUICKSORT (A, 1, 5) => PARTITION (A, 1, 5)

p and r are switched => returns q = 1

QUICKSORT (A, q+1, r) => QUICKSORT (A, 2, 5) => PARTITION (A, 2, 5)

=> returns q = 5

QUICKSORT (A, p, q-1) => QUICKSORT (A, 2, 4) => PARTITION (A, 2, 4)

p and r are switched => returns q = 2

QUICKSORT (A, q+1, r) => QUICKSORT (A, 3, 4) => PARTITION (A, 3, 4)

=> returns q = 4

3/ QuickSort Algorithm Running Time

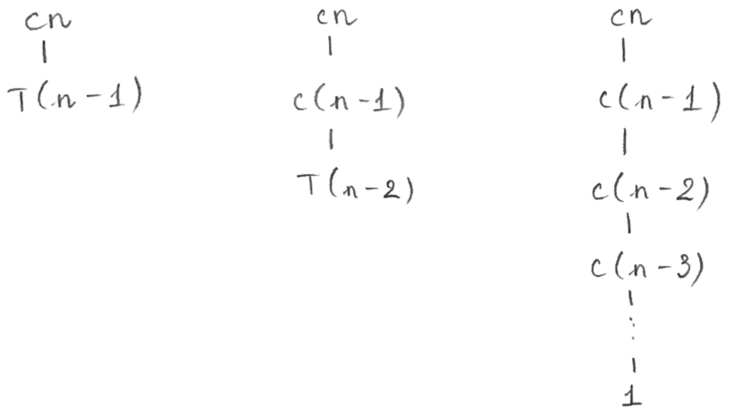
20

QuickSort (A, p, r)	Cost	# of times
1. if $p < r$	$C1$	1
2. $q = \text{PARTITION}(A, p, r)$	$\theta(n)$	1
3. $\text{QUICKSORT}(A, p, q-1)$	$T(0)$	1
4. $\text{QUICKSORT}(A, q+1, r)$	$T(n-1)$	1

$T(n) = C1 + \theta(n) + T(0) + T(n-1)$

$T(n) = T(n-1) + cn$ (Because $\theta(n) = cn$ and $T(0)$ is an empty subarray)

$T(n-1) = T(n-2) + c(n-1)$



$$T(n) = c\,(n + (n-1) + (n-2) + (n-3) + \dots + 1)$$

$$= c\,\frac{n(n+1)}{2}$$

$$= c\,\frac{n^2 + n}{2}$$

$$= \theta(n^2)$$

4/ QuickSort and Substitution

a) $T(n) = T(n-1) + cn$

- Upper bound: $T(n) \leqslant T(n-1) + cn$ (c: positive constant)

Guess: $T(n) = O(n^2)$

$\leqslant dn^2$ (d: positive constant)

$T(n-1) \leqslant d(n-1)^2$

$T(n-1) \leqslant d(n^2 - 2n + 1)$

Substitution: $T(n) \leqslant dn^2 - 2dn + d + cn$

$= dn^2 + (c-2d)n + d$

$\leqslant dn^2$ (if $(c-2d)n + d \leqslant 0$)

$\Rightarrow c - 2d \leqslant 0$

$\Rightarrow d \geqslant c/2$

$T(n) = O(n^2)$ (1)

- Lower bound: $T(n) \geqslant T(n-1) + cn$ (c: positive constant)

Guess: $T(n) = \Omega(n^2)$

$\geqslant d(n^2)$ (d: positive constant)

$T(n-1) \geqslant d(n-1)^2$

$T(n-1) \geqslant d(n^2 - 2n + 1)$

Substitution: $T(n) \geqslant dn^2 - 2dn + d + cn$

$= dn^2 + (c-2d)n + d$

$\geqslant dn^2$ (if $(c-2d)n + d \geqslant 0$)

$\Rightarrow c - 2d \geqslant 0$

$\Rightarrow d \leqslant c/2$

$T(n) = \Omega(n^2)$ (2)

From (1) and (2),

$T(n) = \theta(n^2)$

5/ QuickSort Analysis

15

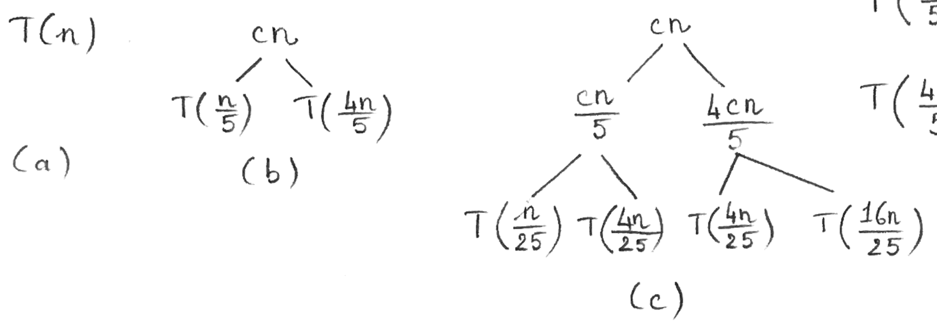
Assume the partitioning algorithm always produces an 80-to-20 proportional split

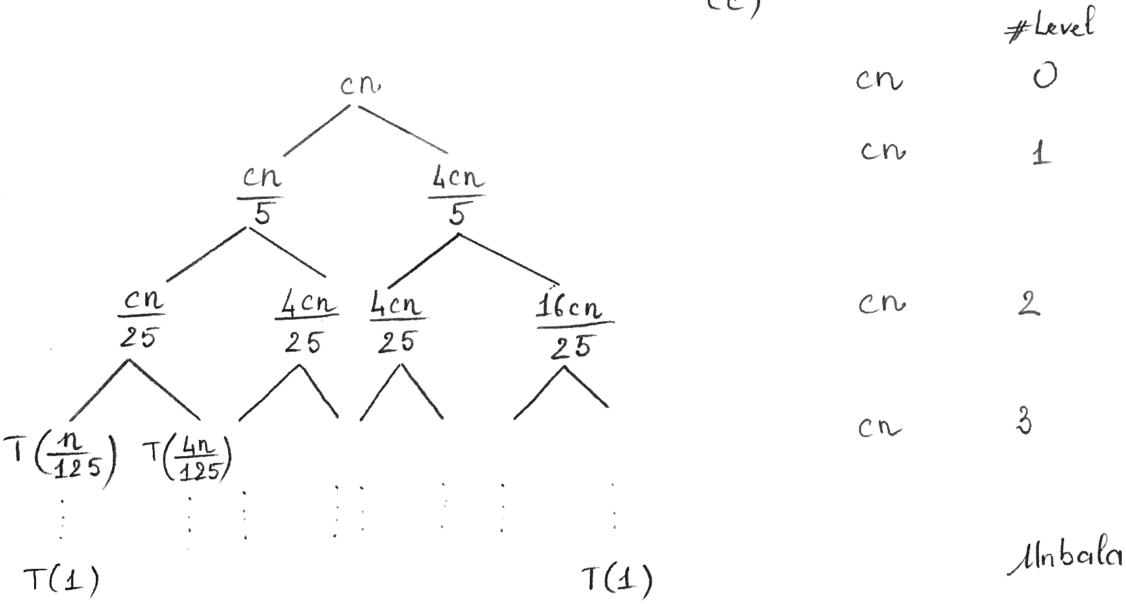
$$T(n) = T\left(\frac{80}{100}n\right) + T\left(\frac{20}{100}n\right) + cn \qquad (c: \text{positive constant})$$

$$= T\left(\frac{n}{5}\right) + T\left(\frac{4n}{5}\right) + cn$$

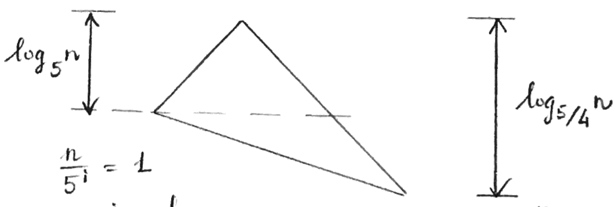
$$T\left(\frac{n}{5}\right) = T\left(\frac{n}{25}\right) + T\left(\frac{4n}{25}\right) + \frac{cn}{5}$$

$$T\left(\frac{4n}{5}\right) = T\left(\frac{4n}{25}\right) + T\left(\frac{16n}{25}\right) + \frac{4cn}{5}$$





Unbalanced tree



$\frac{n}{5^i} = 1$

$\Rightarrow i = \log_5 n$

$\frac{n}{(\frac{5}{4})^i} = 1$

$\Rightarrow i = \log_{5/4} n$

The left most peters out after $\log_5 n$ level

The right most peters out after $\log_{5/4} n$ level

There are $\log_5 n$ full level cost $\log_5 n . cn$

From $\log_5 n$ to $\log_{5/4} n$ cost for each level $\leqslant cn$

$$T(n) \geqslant c \sum_{i=0}^{\log_5 n} n = \Omega(n\lg n)$$

$$T(n) \leqslant c . \sum_{i=0}^{\log_{5/4} n} n = O(n\lg n)$$

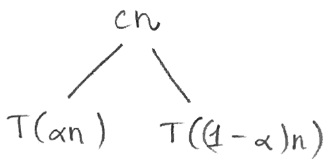
Therefore, $T(n) = \Theta(n\lg n)$

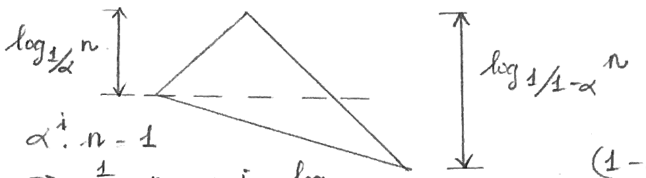
15

6/ QuickSort Analysis (Exercise 7.2-5)

Suppose that the splits are every level of quicksort are in the proportion $1-\alpha$ to α, where $0 \leqslant \alpha \leqslant 1/2$ is a constant.

$$T(n) = T(\alpha n) + T((1-\alpha)n) + cn \qquad (c: \text{positive constant})$$





$\alpha^i . n = 1$

$\Rightarrow \frac{1}{\alpha^i} = n \Rightarrow i = \log_{1/\alpha} n$

$(1-\alpha)^i n = 1$

$\Rightarrow \frac{1}{(1-\alpha)^i} = n$

$\Rightarrow i = \log_{1/1-\alpha} n$

- Minimum depth of a leaf: At the level i:

$$\alpha^i n = 1 \Rightarrow i = \log_\alpha\left(\frac{1}{n}\right) = \frac{\lg(\frac{1}{n})}{\lg(\alpha)} = \frac{\lg(1) - \lg(n)}{\lg(\alpha)} = \frac{-\lg n}{\lg \alpha}$$

- Maximum depth of a leaf: At the level i:

$$(1-\alpha)^i n = 1 \Rightarrow i = \log_{(1-\alpha)}\left(\frac{1}{n}\right) = \frac{\lg(\frac{1}{n})}{\lg(1-\alpha)} = \frac{\lg(1) - \lg(n)}{\lg(1-\alpha)} = \frac{-\lg n}{\lg(1-\alpha)}$$