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: 6 / 12 / 2019

Name (please print): PHONG VO

Signature: [signature]

Due Date: 06-13-2019 (Th), BEFORE the class begins

This assignment covers textbook Chapter 5, 7 and Chapter1~4.

For question 1 and 2: please clearly specify (1) what the indicator random variable is and (2) what does it represent, and (3) show how you use the linearity of expectation and lemma 5.1 to calculate the result. Only have a result can't get the full credits.

1. **Indicator Random Variables** (20 points)
There are n people at a circular table in a restaurant. On the table there are n different appetizers arranged on a big Lazy Susan (which is a turntable or rotating tray placed on a table or countertop to help distribute food). Each person starts eating the appetizer directly in front of him or her. Then a waiter spins the Lazy Susan so that everyone is faced with a random appetizer.

What is the expected number of people that end up with the appetizer that they had originally? You must define necessary random variable and/or indicator random variable clearly.

2. **Indicator Random Variables** (25 points)
Exercise 5.2.5 (P122)

3. Use the randomized quick sort algorithm below to answer question. (15 points)

```
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)
```

Array A contains 1000 distinct elements. After the first call to RANDOMIZED-PARTITION, the return value is 201 (i.e., the pivot location is 201). When call RANDOMIZED-QUICKSORT to sort the second sub-array recursively, what is the expected value returned from RANDOM function?

4. **QuickSort Algorithm** (40 points)
Problem 7.2 (P186)

* Some textbooks contain typos for c, the below is the correct description.

c. Modify the RANDOMIZED-PARTITION procedure to call PARTITION′, and name the new procedure RANDOMIZED-PARTITION′. Then modify the QUICKSORT procedure to produce a procedure QUICKSORT′(A, p, r) that calls RANDOMIZED-PARTITION′ and recurses only on partitions of elements not known to be equal to each other.

Problem 1: Indicator Random Variables

There are n people being seated.

There are also n different appetizers placed in front of each person.

After the first bite, the big Lazy Susan is rotated.

- ⇨ Sample space S = n
- ⇨ Event {people meet their original appetizers again} X = 1
- ⇨ Pr{X} = $\frac{1}{n}$
- ⇨ E[X_i] = $\frac{1}{n}$
- ⇨ E[X] $= \sum_{i=1}^{n} E[X_i]$

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

$$= \frac{1}{n} + \frac{1}{n} + \ldots + \frac{1}{n} \quad \text{(n times)}$$

$$= n * \frac{1}{n} = 1$$

- ⇨ E[X] = 1

Problem 2: Indicator Random Variables

Exercise 5.2.5 (p 122)

Let X_{ij} be an indicator random variable where (i,j) is called an *inversion* of A.

$X_{ij} = I\{A[i] > A[j]\}$

$= \begin{cases} 1 \ (if\ A[i] > A[j]) \\ 0 \ (elsewhere) \end{cases}$ for $1 \le i \le j \le n$

⇨ $\Pr\{X_{ij}=1\} = \frac{1}{2}$

⇨ $E[X_{ij}] = \frac{1}{2}$ (by Lemma 5.1)

$E[X] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E[X_{ij}]$

$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{2}$

$= \frac{1}{2} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} 1$

$= \frac{1}{2} \sum_{i=1}^{n-1} (n-i)$

$= \frac{1}{2}$ [(n-1) + (n-2) + (n-3) + . . + (n-n+1)]

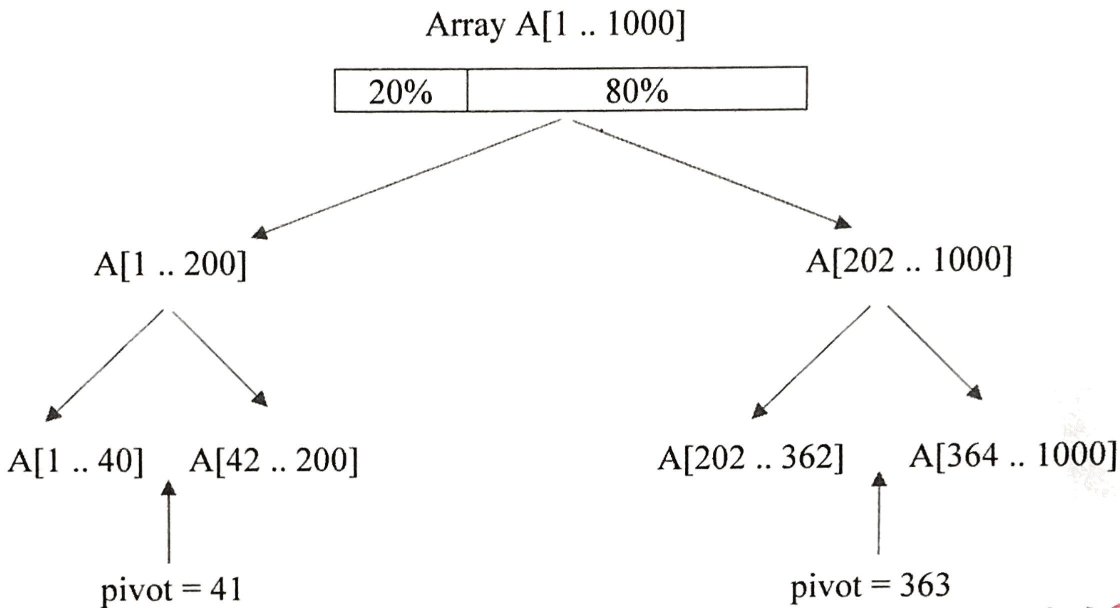
$= \frac{1}{2}$ [(n-1) + (n-2) + (n-3) + . . + 1]

$= \frac{1}{2} [\frac{n(n-1)}{2}]$

⇨ $E[X] = \frac{n(n-1)}{4}$

Problem 3: Randomized Quick Sort

The size of the array is 1,000 which is divided into 2 sub-arrays at rate 1:4. The return value after RANDOMIZED has been called is 201 means the pivot of its current iteration is 201, the first array has range [1 .. 200] and the second array has range [202 .. 1000].



Total num of possible value : 1000-202+1=799

Therefore, after the second sub-array has been sorted by calling RANDOMIZED-QUICKSORT, the expected values from RANDOM function are 41 and 363. 601

-15.

Problem 4: Quicksort Algorithm (Exercise 7-2 p186)

a/

If all elements' values are the same, PARTITION returns the value of q and r in the same value as the array A[p .. q-1] does.

The recurrence is:

$T(n) = T(n-1) + T(0) + cn$

$= T(n-1) + cn$

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

b/

```
PARTITION'(A, p, r)
    x = A[p]
    i = h = p
    for j = p+1 to r
        if A[j] < x
            y = A[j]
            A[j] = A[h+1]
            A[h+1] = A[i]
            A[i] = y
            ++i
            ++h
        else if A[j] == x
            exchange A[h+1] with A[j]
            ++h
    return (i, h)
```

c/

RANDOMIZED-PARTITION' is the same as RANDOMIZED-PARTITION but calling PARTITION' instead of PARTITION.

HW5

```
QUICKSORT'(A, p, r)
    if q < r
        (q, t) = RANDOMIZED-PARTITION'(A, q, r)
        QUICKSORT' (A, p, q-1)
        QUICKSORT' (A, t+1, r)
```

d/ Put elements which are same value as pivot's in the same partition.

This makes problem sizes of QUICKSORT's no longer than those of the original QUICKSORT when all elements are distinct, and even with equal number of elements.

85/100