Due Date: 04-01-2019 (M), 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** (15 points)
   Exercise 5.2.4 (P122)

2. **Indicator Random Variables** (30 points)
   Exercise 5.2.5 (P122)

3. **Uniform Random Permutation** (15 points)
   Exercise 5.3.3 (P129)

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.

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: 04/01/2019

Name (please print): DANGNHI NGO

Signature:

98/100

15 **1/ Indicator Random Variables**

Exercise 5.2.4

For n >= i >= 1, let

X_i= I {customer i gets his hat back}

Let random variable X be the number of customers getting hat back.

X= $\sum_{i=1}^{n} X_i$

Pr {X_i}= $\frac{1}{n}$ => E [X_i] = $\frac{1}{n}$ (lemma)

E[X] = E [$X_1+X_2+\ldots+X_n$] = $E[\sum_{i=1}^{n} X_i] = \sum_{i=1}^{n} E[X_i] = \sum_{i=1}^{n} \frac{1}{n} = 1$

30 **2/ Indicator Random Variables**

Exercise 5.2.5

Let X_{ij} be an indicator random variable

For (i ,j) ∊ inversions of A,

X_{ij} = I { the first is greater than the second}

= I $\{ {}^{1}_{0} \}$

⇨ Pr {X_{ij}=1} = $\frac{1}{2}$

⇨ E [X_{ij}] = $\frac{1}{2}$ (lemma)

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

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

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

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

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

$= \frac{n(n-1)}{4}$

15

3/ Uniform Random Permutation
Exercise 5.3.3
No.
Because in each n iteration, the algorithm chooses the index i independently and uniformly at random {1….n}
Therefore, there are n^n different possible sequences, each has Pr = $\frac{1}{n^n}$.
There are n! distinct permutations. To get a uniform distribution over permutations, each has Pr

$= \frac{1}{n!}$.

This we have $\frac{k}{n^n} = \frac{1}{n!} \Leftrightarrow n^n =$ kn! (k is an integer), thus not possible because n^n is not divisible by

n! (except for n=1 or n=2)

38 **4/ QuickSort Algorithm**

Problem 7.2

a/ If elements are equal, when PARTITION returns, q is equal to r and all elements in

A [p…q-1] are equal.

The recurrence is:

T(n)= T(n-1) + T(0) + θ(n)

T(n) = θ (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 = i+1

        h = h+1

    else if A[j] = x

        exchange A[h+1] with A[j]

        h = h+1

return (i,h)
```

c/ RANDOMIZED – PARTITION′ is the same as RANDOMIZED – PARTITION except for calling to PARTITION being changed to PARTITION.

QUICKSORT′ (A,p,r)

```
1. If q<r
2.     (q,t) = RANDOMIZED – PARTITION′ (A,q,r)
3.     QUICKSORT′ (A,p, q-1)
4.     QUICKSORT′ ( A,t +1, r)
```

d/ Put elements which are equal to the pivot in the same partition as the pivot.

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