1-3. Victor Espaillat

X_i : Indicator random variable associated with the event in which the i^{th} person ends up with the same appetizer.

$$X_i = I\{\text{the } i^{\text{th}} \text{ person gets the same appetizer}\} = \begin{cases} 1 & \text{if person } i \text{ gets same appetizer} \\ 0 & \text{if person } i \text{ does NOT} \end{cases}$$

X : number of people that end up with the same appetizer.

$$X = X_1 + X_2 + \cdots + X_n = \sum_{i=1}^{n} X_i$$

By Lemma 5.1,
$E[X_i] = \Pr\{\text{the } i^{\text{th}} \text{ person gets the same appetizer}\}$

Take one person. Once the tray is spun, that person could end up with one of n possible appetizers. Assuming the tray is equally likely to stop on any appertizer, the probability for that person to end up with a particular one (in this case the one they started with) is simply $1/n$.

$\therefore E[X_i] = 1/n$

$$\begin{aligned} E[X] &= E\left[\sum_{i=1}^{n} X_i\right] \\ &= \sum_{i=1}^{n} E[X_i] \quad \text{(by linearity of expectation)} \\ &= \sum_{i=1}^{n} \frac{1}{n} \\ &= n\left(\frac{1}{n}\right) = 1 \end{aligned}$$

X_{ij} : Indicator random variable associated with the event that the pair (i, j) is an inversion of A.

$$X_{ij} = I\left\{ \text{the pair } (i,j) \text{ is an inversion of A} \right\} = \begin{cases} 1 & \text{if } A[i] > A[j] \\ 0 & \text{if } A[i] \le A[j] \end{cases}$$

$$\begin{aligned} E[X_{ij}] &= \Pr\{X_{ij}\} \\ &= 1 \times \Pr\{A[i] > A[j]\} + 0 \times \Pr\{A[i] \le A[j]\} \\ &= \Pr\{A[i] > A[j]\} \\ &= 1/2 \end{aligned}$$

X : The number of inversions of A.

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

$$= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{1}{2} = \sum_{i=1}^{n-1}\left[\frac{1}{2}\sum_{j=i+1}^{n} 1\right]$$

$$= \sum_{i=1}^{n-1}\left[\frac{1}{2}(n-i)\right] = \frac{1}{2}\sum_{i=1}^{n-1}(n-i)$$

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

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

When RANDOMIZED-QUICKSORT is called to sort the the <u>second</u> sub-array recursively, it will be called with parameters $(A, (q+1) = 202,\ r = 1000)$.

Then RANDOMIZED-PARTITION will be called with $(A, p = 202, r = 1000)$ and consequently RANDOM will be called with $(p = 202, r = 1000)$.

RANDOM(202,1000) returns an integer ranging from [202,1000].
There are $(1000 - 202 + 1) = 799$ possible values. Each is equally likely to be returned,
$\therefore \Pr\{i\} = 1/799$

X : value returned by RANDOM(202, 1000)

$$E[X] = \left(202 \cdot \frac{1}{799}\right) + \left(203 \cdot \frac{1}{799}\right) + \left(204 \cdot \frac{1}{799}\right) + \cdots + \left(1000 \cdot \frac{1}{799}\right)$$

$$= \frac{1}{799} \sum_{i=202}^{1000} i$$

$$= 601$$

4. Phong Vo

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:

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

$\quad = T(n-1) + cn$

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

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.

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 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.