1. By Duyen Tran

X = # of customers that get their own hat back

x_i = Indicator Random Variable associated with the event that customer i gets their own hat back

$x_i = I\{\text{customer } i \text{ gets their hat back}\} = \begin{cases} 1 & \text{if they do} \\ 0 & \text{if they don't} \end{cases}$

$$E[x_i] = Pr(x_i \text{ gets their hat back})$$
$$= \frac{1}{n}$$
$$E[X] = E\left[\sum_{i=1}^{n} x_i\right]$$
$$= \sum_{i=1}^{n} E[x_i]$$
$$= \sum_{i=1}^{n} \frac{1}{n}$$
$$= \frac{n}{n} = \underline{1}$$

the expected # of customers who will get their own hat back is 1

2. By James Tan

Indicator Random Variable: The variable that $A[i] > A[j]$; 1 if true, 0 if not ($A[i] < A[j]$). Since the permutation is uniform random, we know nothing about the array except that all the numbers are distinct. In that case, the probability of the variable being true is 1/2 and also 1/2 for false.

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

chance of variable holding true (indicator random variable)

from 1 to n-1 in array because last element cannot be compared to itself

comparison of next element to first element all the way to n

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E[X_{ij}] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{2} = \sum_{i=1}^{n-1} \cdot \left(\frac{1}{2}(n-(i+1)+1)\right)$$

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

$1 + 2 + \dots + n-2 + n-1$

$n-1 + n-2 + \dots + 2 + 1$

$$2\Sigma = (n-1)n$$

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

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

$$= \frac{1}{2}\left[\frac{1}{2}n^2 - \frac{1}{2}n\right] = \frac{1}{4}[n^2 - n] = \frac{n(n-1)}{4}$$

3. By Bonnie Liu

PERMUTE-WITH-ALL(A)

```
n = A.length
for i = 1 to n
    swap A[i] with A[RANDOM(1, n)]
```

Does this code produce a uniform random permutation? Why or why not?

Let $n = 3$, then there are 27 possible outcomes of calling permute - will-all
Assume pemute-with-all produce a uniform random permutation.
then $3! = 3 \times 2 \times 1 = 6$, for each permutation would occur $\frac{1}{6}$ times.

So each Pemutation have to occur i times.
but no integer i can satisfies $\frac{i}{27} = \frac{1}{6}$.

from above we know this code does not produce a uniform random permutation.

4. By Steven Gaudet

4. a. Partition will return $q = r$ so, with all elements in the array being equal, the recurrence will be $T(n) = T(n-1) + \Theta(n) + T(0)$ for the running time.
$\therefore T(n) = \Theta(n^2)$

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 = 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(A, p, r)
    i = Random(p, r)
    exchange A[r] with A[i]
    return Partition'(A, p, r)
```

```
Quicksort'(A, p, r)
    if q < r
        (q, i) = Randomized-Partition'(A, q, r)
        Quicksort'(A, p, q-1)
        Quicksort'(A, i+1, r)
```

d. Any elements that're equal to the pivot should be put in the same partition as pivot because we don't recurse on those elements. So the subproblem size with Quicksort' isn't larger than the subproblem size of Quicksort even with equal elements.