Worst-case: \( n \) hires

**Average-case** analysis using **indicator variables**: \( \ln n + O(1) \) hires

- assumes each of the \( n! \) permutations of rankings is equally likely

**Expected** analysis: average run-time, **regardless of the input**

achieved by **randomizing** the input
A randomized algorithm for the hiring problem

```
RANDOMIZED-HIRE-ASSISTANT(n)
1  randomly permute the list of candidates
2  best = 0  // candidate 0 is a least-qualified dummy candidate
3  for i = 1 to n
4      interview candidate i
5      if candidate i is better than candidate best
6         best = i
7      hire candidate i
```
Randomly permuting arrays

PERMUTE-BY-SORTING ($A$)

1. $n = A.length$
2. let $P[1..n]$ be a new array
3. for $i = 1$ to $n$
4.   $P[i] = \text{RANDOM}(1, n^3)$
5. sort $A$, using $P$ as sort keys

Makes it very likely that the $n$ values will all be unique
Randomly permuting arrays

**RANDOMIZE-IN-PLACE**(A)

1. $n = A.length$
2. **for** $i = 1$ **to** $n$
3. swap $A[i]$ with $A[RANDOM(i, n)]$