Randomness in Computing

CS 537

LECTURE 6

Last time

- Conditional expectation
- Branching process
- Bernoulli, binomial, and geometric RVs

Today

- Coupon collector problem
- Randomized quicksort

CS 537

Geometric random variables

The geometric distribution with parameter p, denoted Geom(p), is the distribution of the number of tosses of a coin with bias p until it shows HEADS.

- Lemma. The probability distribution of $X \sim$Geom(p) is
$$\Pr[X = n] = (1-p)^{n-1}p$$
for all $n = 1,2, \ldots$.
- Lemma. The expectation of $X \sim$Geom(p) is
$$\mathbb{E}[X] = 1/p.$$

Exercise

- What is the distribution of the number of rolls of a die until you see a 6?
- What is the expected number of rolls until you see a 6?

Exercise

- You roll a die until you see a 6. Let X be the number of 1s you roll. Compute $\mathbb{E}[X]$.
- Hint: Let N be the number of rolls.
- Solution: We will condition on N:

$$\mathbb{E}[X] = \mathbb{E}\big[\mathbb{E}[X|N]\big].$$

$$\mathbb{E}[X|N = n] = \frac{n-1}{5}$$

$$N \sim Geom(1/6)$$

$$\mathbb{E}[X] = \mathbb{E}\big[\mathbb{E}[X|N]\big] = \mathbb{E}\left[\frac{N-1}{5}\right] = \frac{6-1}{5} = 1.$$

Exercise

- You roll a die until you see a 6. Let S be the sum of the rolls. Compute $\mathbb{E}[S]$.
- Hint: Let N be the number of rolls.
- Solution: We will condition on N:

$$\mathbb{E}[S] = \mathbb{E}\big[\mathbb{E}[S|N]\big].$$

$$\mathbb{E}[S|N=n] = 3(n-1)+6 = 3n+3$$

$$N \sim Geom(1/6)$$

$$\mathbb{E}[S] = \mathbb{E}\big[\mathbb{E}[S|N]\big] = \mathbb{E}[3N+3]$$

$$= 3\cdot 6+3 = 21$$

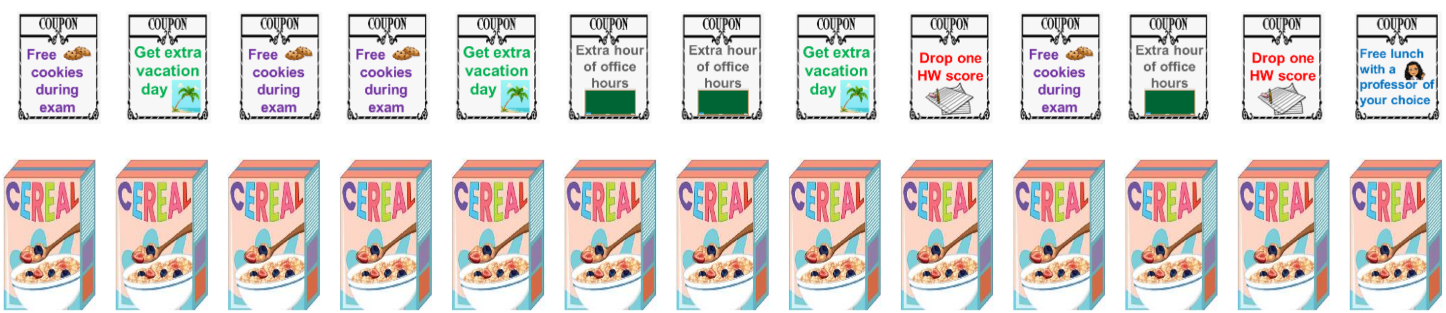
Coupon Collector's Problem

- There are n coupons

- Each cereal box has 1 coupon chosen uniformly and independently at random
- What is the expected number of boxes you need to buy to collect all n coupons?
  - X = the number of boxes bought to collect at least one copy of each coupon
  - Find $\mathbb{E}(X)$

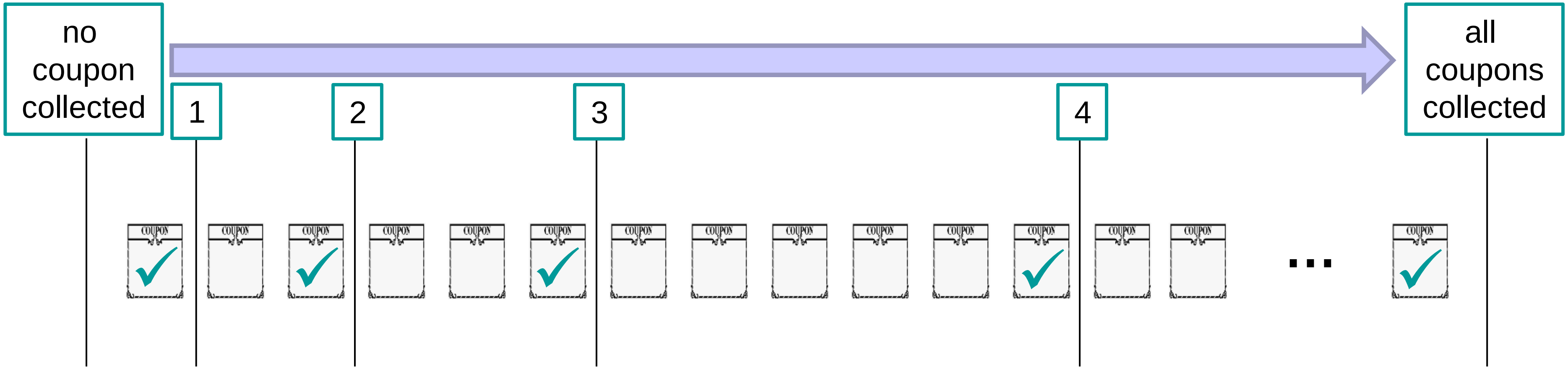
Example

$n = 5$



$X = 13$

Breaking it Down into Phases



Coupon Collector's Problems

- There are n coupons.
- Each cereal box has 1 coupon chosen u.i.r.
- X = # of boxes bought until at least one copy of each coupon is obtained.
- Find $\mathbb{E}[X]$.

Solution: Let $X_i =$ # of boxes bought while you had exactly $i - 1$ different coupons.

Then $X = X_1 + X_2 + \cdots + X_n$

$X_i \sim$

$\mathbb{E}[X_i] =$

$\mathbb{E}[X] = \sum_{i=1}^{n}$

Lemma. $\ln n \leq H(n) \leq \ln n + 1$, where $H(n) = \sum_{i=1}^{n} \frac{1}{i}$

Intuition:

CS 537

Duration of a Random Experiment

Let random variable X denote the duration of a random experiment.

Approach to calculate $\mathbb{E}(X)$

- Carefully **partition** the experiment into phases.
- Calculate the expected duration of each phase.
- Use linearity of expectation to calculate $\mathbb{E}(X)$.

Application of Coupon Collector's

Coupon Collector's Problem has many applications in CS

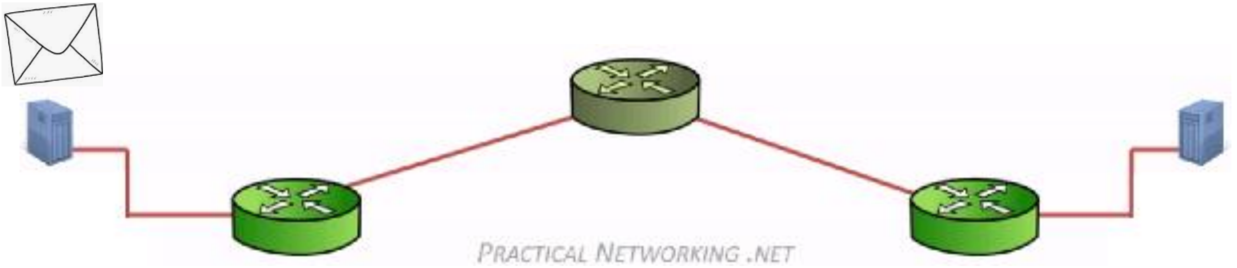
Example:

- Packets passing along a fixed path of n routers.
- Designation host wants to collect names of the routers, but each packet has only space for one name. and a counter.

Idea: Sample the name of a uniformly random router.

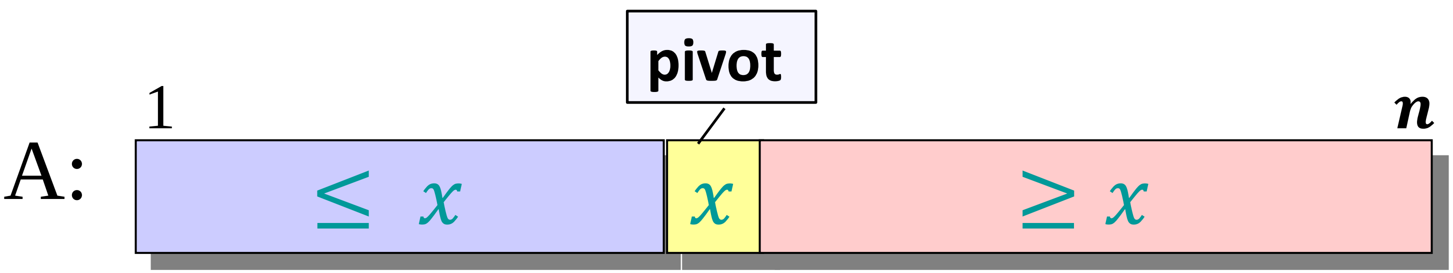
Reservoir sampling: k-th router replaces previous router in the header w.p. $1/k$

- What is the expected number of packets the designation host needs to see in order to collect the names of all routers?

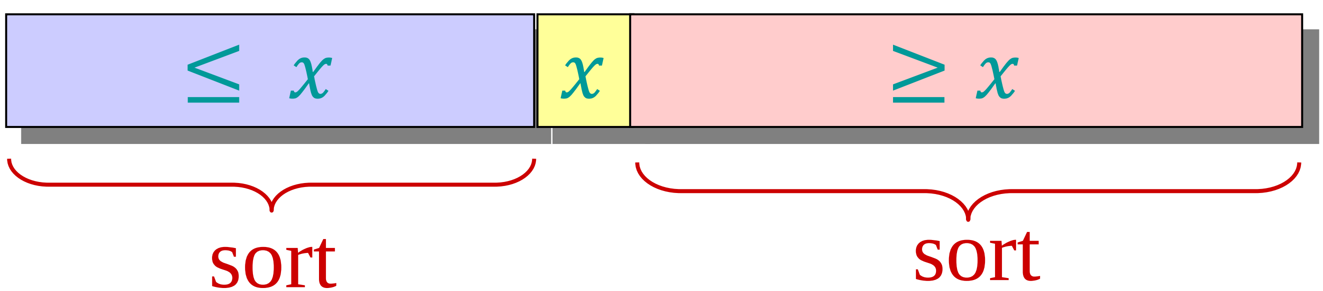


Quicksort: divide and conquer

- Find a *pivot* element
- Divide: Find the correct position of the pivot by comparing it to all elements.



- Conquer: Recursively sort the two parts, resulting from removing the pivot.



Quicksort

QuickSort(array A, positive integers ℓ, r)

1. **if** $\ell < r$
2. **then** $p \leftarrow$ Partition(A, ℓ, r)
3. QuickSort (A, $\ell, p - 1$)
4. QuickSort (A, $p + 1, r$)

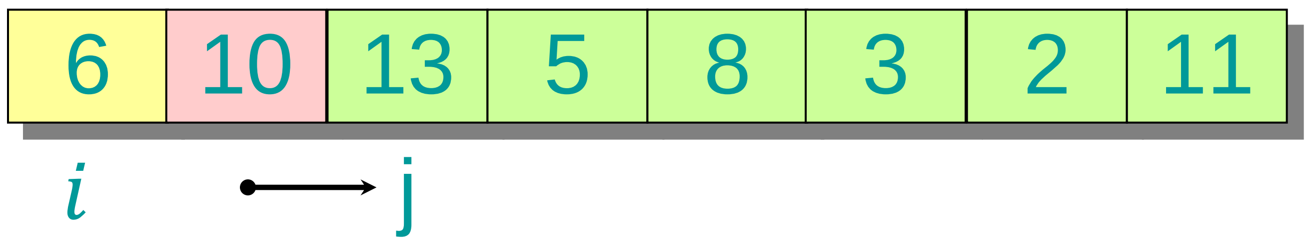
Initial call: QuickSort (A, 1, n)

Example of partitioning

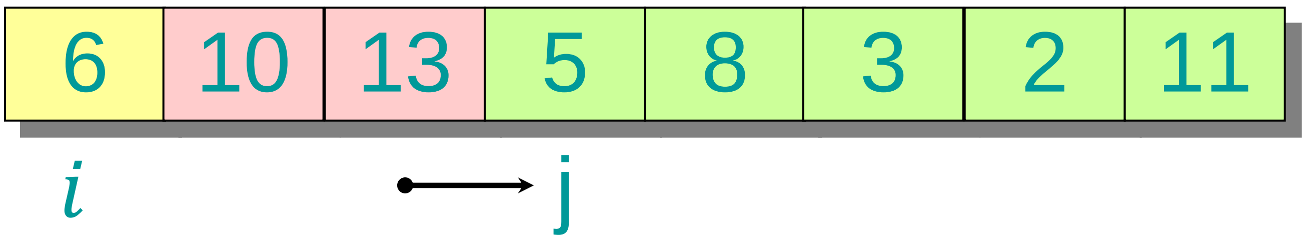
6	10	13	5	8	3	2	11
i	j						

CS 537

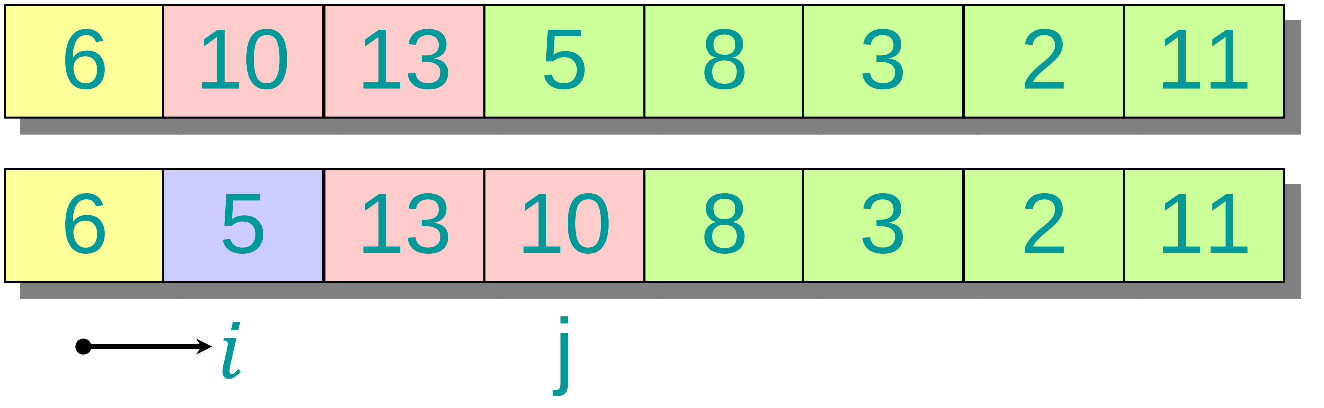
Example of partitioning



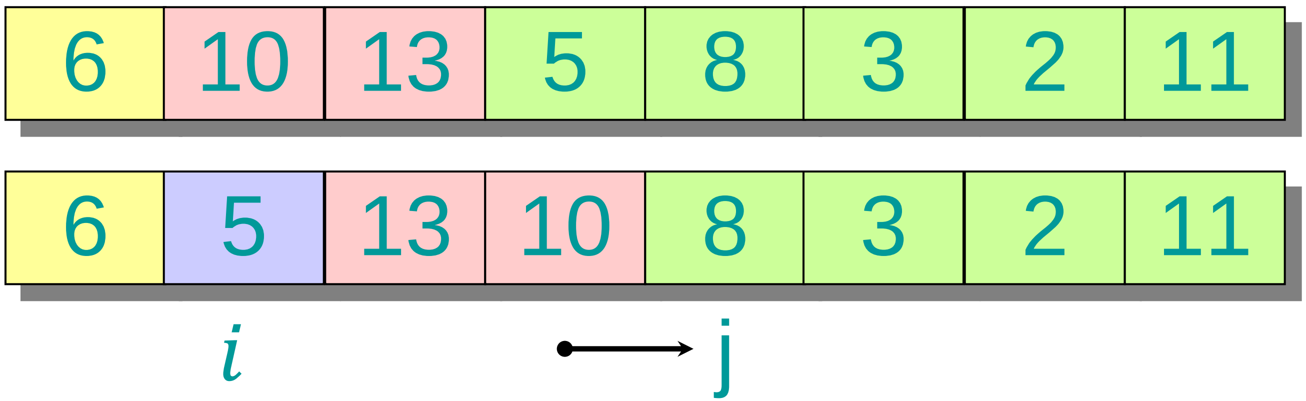
Example of partitioning



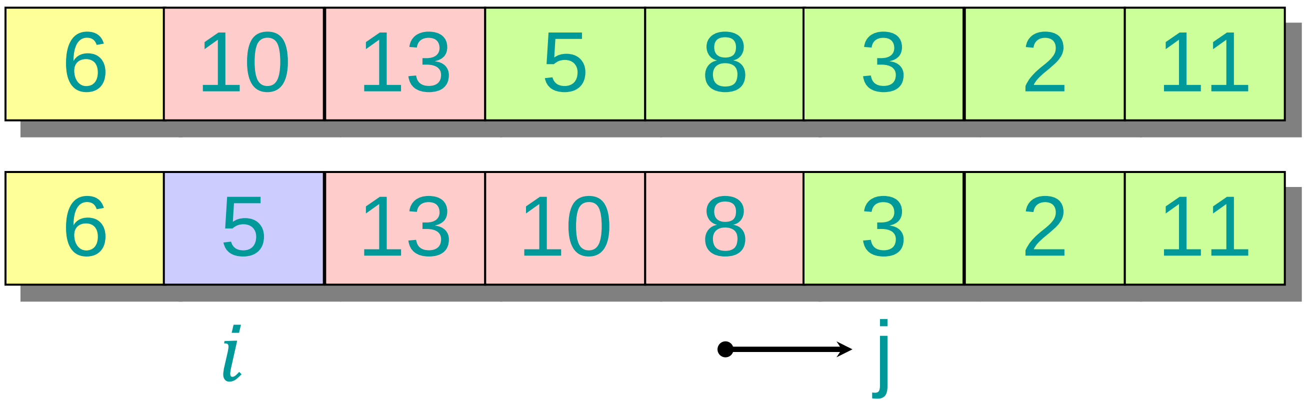
Example of partitioning



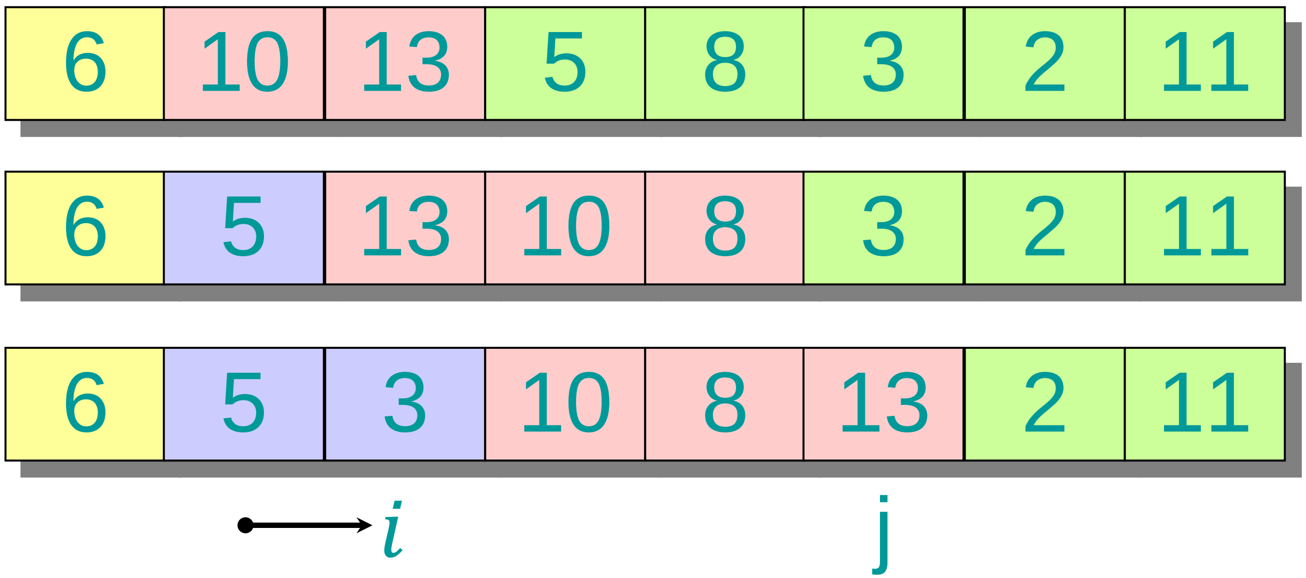
Example of partitioning



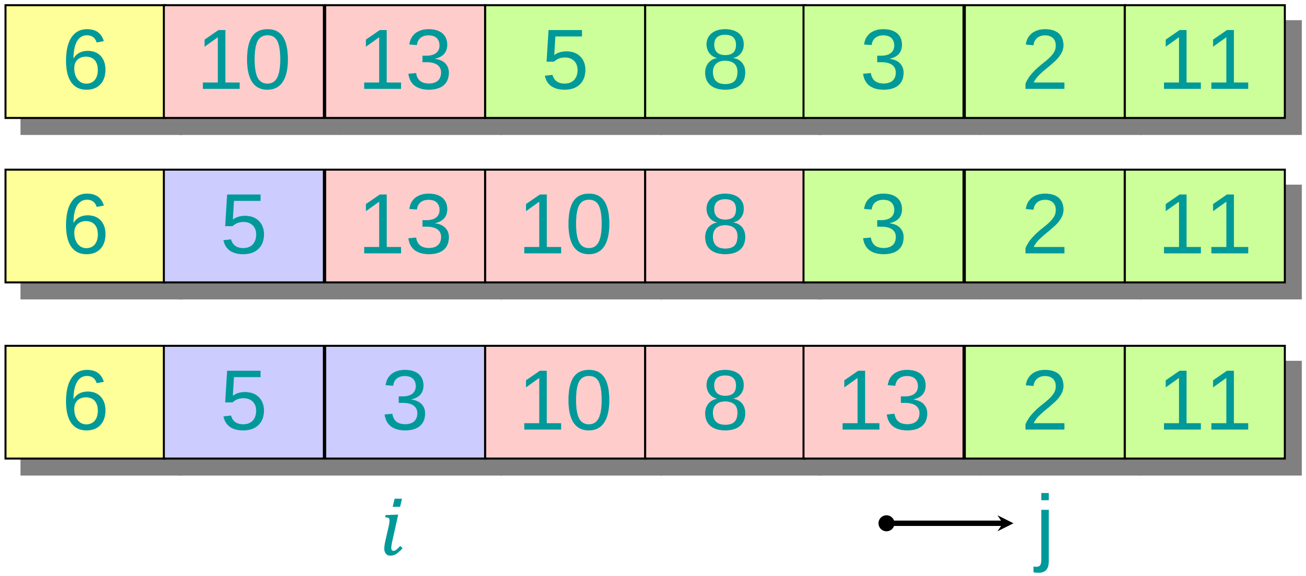
Example of partitioning



Example of partitioning



Example of partitioning



Example of partitioning

6	10	13	5	8	3	2	11

6	5	13	10	8	3	2	11

6	5	3	10	8	13	2	11

6	5	3	2	8	13	10	11

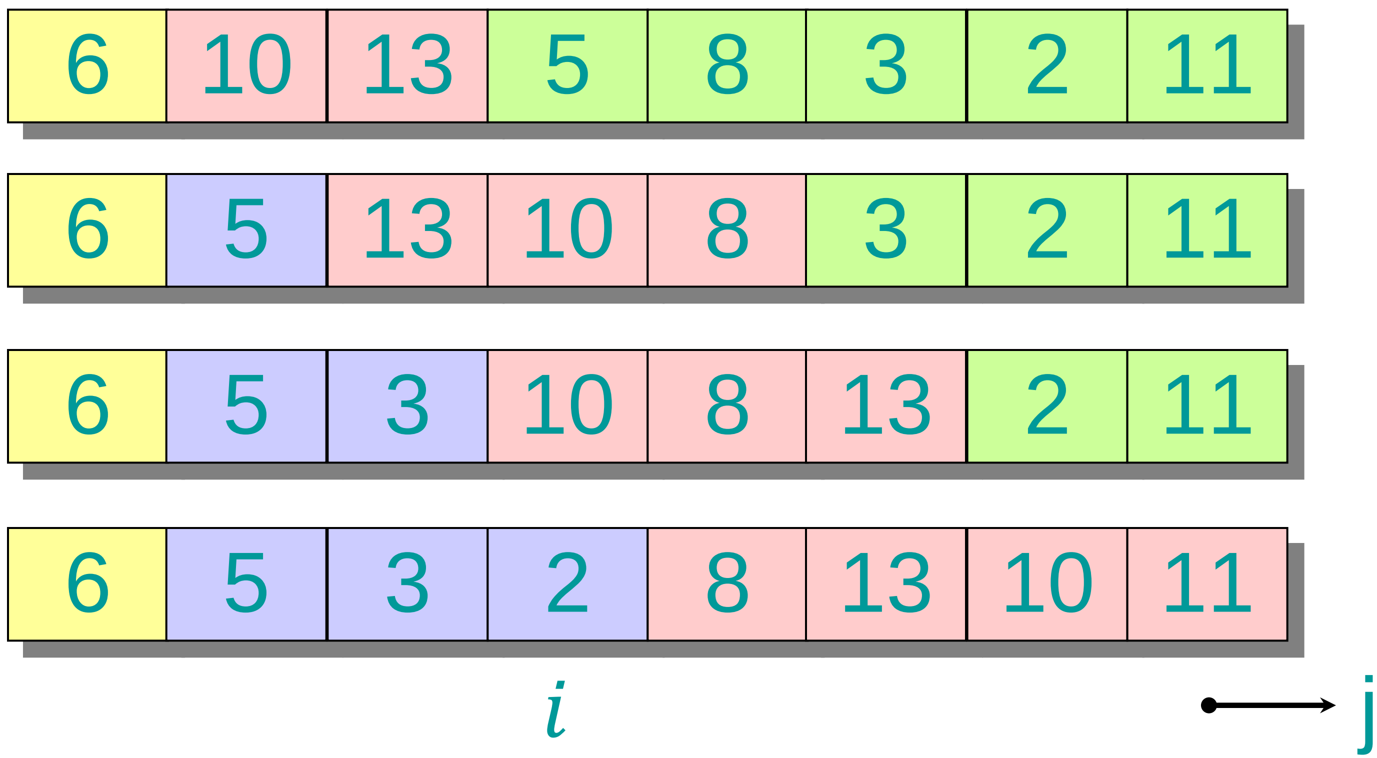
→ i $\qquad$ j

Example of partitioning

6	10	13	5	8	3	2	11
6	5	13	10	8	3	2	11
6	5	3	10	8	13	2	11
6	5	3	2	8	13	10	11

i → j

Example of partitioning



Example of partitioning

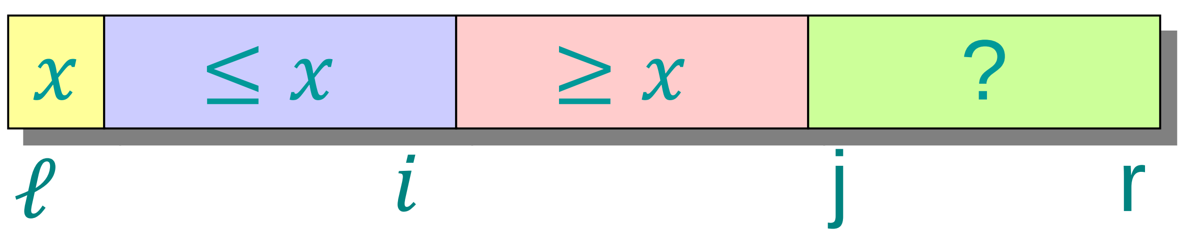
6	10	13	5	8	3	2	11
6	5	13	10	8	3	2	11
6	5	3	10	8	13	2	11
6	5	3	2	8	13	10	11
2	5	3	6	8	13	10	11

i

Partitioning algorithm

Partition (array A, positive integers ℓ, r)

```
x ← A[ℓ]        // A[ℓ] becomes the pivot
i ← ℓ
for j = ℓ + 1 to r
    if A[j] < x
        then i ← i + 1
             SWAP(A[i], A[j])
SWAP(A[ℓ], A[i])
return i
```



CS 537

Exercise

How many comparisons does Quicksort perform on sorted array?

Answer:

$$(n-1)+(n-2)+\cdots+2+1=\frac{n(n-1)}{2}=\Omega(n^2)$$

How many comparisons does Quicksort perform if, in every iteration, the pivot splits the array into two halves?

Answer:

Let $C(n)$ be the number of comparisons performed on an array with n elements.

$$C(n)=\Theta(n\log n)$$

Randomized Quicksort

BIG IDEA:

Partition around a ***random*** element.

- Analysis is similar when the input arrives in random order.
- But randomness in the input is unreliable.
- Rely instead on random number generator.

CS 537 Analysis of Randomized Quicksort

Theorem. If Quicksort chooses each pivot uniformly and independently at random from all possibilities then, for any input, the expected number of comparisons is

$$2n \ln n + O(n).$$

Proof (with an assumption that all elements are distinct):

- Let X be the R.V. for the # of comparisons.
- Let $x_1, x_2, \ldots, x_n$ be the input values.
- Let $y_1, y_2, \ldots, y_n$ be the input values sorted in increasing order.
- For $i, j \in [n], i < j$, let X_{ij} be the indicator R.V. for the event that y_i and y_j are compared by the algorithm.

$$X = \sum_{i,j \in [n]:\, i<j} X_{ij} \text{ and, by linearity of expectation, } \mathbb{E}[X] = \sum_{i,j [n]: i<j} \mathbb{E}\left[X_{ij}\right]$$

CS 537 Analysis of Randomized Quicksort

Theorem. The expected number of comparisons is $2n \ln n + O(n)$.

Proof (continued):

- Let $y_1, y_2, \dots, y_n$ be the input values sorted in increasing order.
- For $i, j \in [n], i < j$, let X_{ij} be the indicator R.V. for the event that y_i and y_j are compared by the algorithm.
- $\mathbb{E}[X_{ij}] = \Pr[X_{ij} = 1]$
- Important idea: y_i and y_j are compared iff either y_i or y_j is the first pivot chosen from $Y_{ij} = \{y_i, \dots, y_j\}$
- The first time a pivot is chosen from Y_{ij}, it is equally likely to be any of $j - i + 1$ elements of Y_{ij}.

Analysis of Randomized Quicksort

Theorem. The expected number of comparisons is $2n \ln n + O(n)$.

Proof (continued):

$$\Pr[\mathrm{X_{ij}} = 1] = \Pr[y_i \text{ or } y_j \text{ is the first pivot chosen from } Y_{ij}]$$

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

$$\mathbb{E}[X] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1} = \sum_{i=1}^{n-1} \sum_{k=2}^{n-i+1} \frac{2}{k}$$

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

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