Lecture 25

Last time
- Exponential distribution
- Poisson process

Today
- Poisson distribution
- Use of probability in algorithms

Reminder
- HW 12 is due Thursday
Definition: A **Poisson Process** is a sequence of arrivals over time, where:

1) The mean arrival rate

\[ \lambda = \frac{\text{Mean Number of Arrivals}}{\text{Unit Time}} \]

is constant over all time, for any unit interval \([ t .. t+1 )\)

2) The number of arrivals in two non-overlapping intervals is independent; and

3) The probability of two events occurring at the same time is 0.
Poisson Random Variables

- Consider a Poisson Process
- Fix the unit time interval (e.g., 1 second or 1 year)
- Supposed the mean number of arrivals in a unit interval is \( \lambda \)
- Let \( X \) be the number of arrivals in a unit interval. (It has the same distribution for each unit interval.)
- Then we call \( X \) a Poisson Random Variable with rate parameter \( \lambda \), denoted \( X \sim \text{Poisson}(\lambda) \)

\[
Rng(X) = \{0, 1, 2, \ldots\}
\]

\[
f(k) = \frac{\lambda^k e^{-\lambda}}{k!}
\]

\[
E(X) = \lambda
\]

\[
Var(X) = \lambda
\]
Assume that arrivals of email in my Inbox are a Poisson Process with rate $\lambda = 10$ messages per hour.

What is the probability that I get exactly 10 emails in the next hour?

$$Pr(X = 10) = f_X(10) = \frac{\lambda^k e^{-\lambda}}{k!} = \frac{10^{10} e^{-10}}{10!} \approx 0.1251$$
Assume that arrivals of email in my Inbox are a Poisson Process with rate $\lambda = 10$ messages per hour.

What is the probability that I get between 5 and 15 emails (inclusive) in the next hour?

To compute the CDF or ranges, add together all the individual values:

$$
\Pr(5 \leq X \leq 15) = \sum_{k=5}^{15} \frac{e^{-10}10^k}{k!} \approx 0.922
$$

$$
X \sim \text{Poi}(10)
$$

$$
R_X = \{ 0, 1, 2, 3, \ldots \}
$$

$$
P(X = k) = f_X(k) = \frac{e^{-10}10^k}{k!}
$$
Let $Y$ be the time until the first arrival.

The arrivals are independent, so at any time $t$, probabilistically the Poisson process starts all over again (the events don’t remember the past!):

$Y = “\text{the interarrival time between any two events}”$

What is the distribution of $Y$?
Let $Y$ be the time until the first arrival.
Probability is used in the design and analysis of algorithms

1. to analyze deterministic algorithms when inputs come from a specified distribution
2. to design and analyze algorithms that use randomness
An Application: Sorting

How to organize your bookshelf?

- We will show how to do it really quickly when your input comes from a uniform distribution.
The sorting problem

**Input:** n numbers $a_1, a_2, \ldots, a_n$

**Output:** the same numbers in the sorted order:

$$b_1 \leq b_2 \leq \cdots \leq b_n$$

**Example:**

**Input:** 8 2 4 9 3 6
**Output:** 2 3 4 6 8 9
Insertion sort: example

8 2 4 9 3 6
Insertion sort: example

8 2 4 9 3 6
Insertion sort: example

8 2 4 9 3 6
2 8 4 9 3 6
Insertion sort: example
### Insertion Sort: Example

<table>
<thead>
<tr>
<th>8</th>
<th>2</th>
<th>4</th>
<th>9</th>
<th>3</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>8</td>
<td>4</td>
<td>9</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>8</td>
<td>9</td>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>
Insertion sort: example

8 2 4 9 3 6
2 8 4 9 3 6
2 4 8 9 3 6
Insertion sort: example

8 2 4 9 3 6
2 8 4 9 3 6
2 4 8 9 3 6
2 4 8 9 3 6
Insertion sort: example

8 2 4 9 3 6
2 8 4 9 3 6
2 4 8 9 3 6
2 4 8 9 3 6
Insertion sort: example

<table>
<thead>
<tr>
<th>8</th>
<th>2</th>
<th>4</th>
<th>9</th>
<th>3</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>8</td>
<td>4</td>
<td>9</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>8</td>
<td>9</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>8</td>
<td>9</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>8</td>
<td>9</td>
<td>6</td>
</tr>
</tbody>
</table>
Insertion sort: example
Insertion sort: example

8  2  4  9  3  6
2  8  4  9  3  6
2  4  8  9  3  6
2  4  8  9  3  6
2  4  8  9  3  6
2  3  4  8  9  6
2  3  4  6  8  9
done
Insertion Sort: Running Time

We use:

• 1 step to insert the second element
• 2 steps to insert the third element
• ...
• \( n - 1 \) steps to insert the \( n^{th} \) element

Total: \( \frac{4}{26} \) steps
Bucket Sort

- We can sort much faster when the range of numbers is small
- **Example:** Sorting solutions to one HW problem by score (1 to 10)
- **Input:** $n$ integers from range $\{1, \ldots, r\}$

1. Put items with the same value into the same bucket.
2. Keep a linked list for each bucket.
3. Make a pass over the list and put each element in the right bucket
4. Concatenate the lists.

- If $r \leq n$, we can sort in $O(n)$ time
Bucket Sort

- **Given:** $n$ integers from range $[r]$.

- What if $r > n$? (Suppose for simplicity that $n$ divides $r$.)

**Theorem**

If $n$ integers are chosen uniformly and independently from range $\{1, ..., r\}$, they can be sorted in expected time $O(n)$.

- Expectation is over randomness in the choice of integers: Bucket Sort is deterministic (that is, it does not use randomness).
Bucket Sort

- **Idea:** Break the range into $n$ buckets. The expected # of elements in each bucket is 1. We can easily sort all buckets (using Insertion Sort).

Algorithm. Input: integers $a_1, \ldots, a_n$

1. Make linked lists for buckets $B_1, \ldots, B_n$.
2. For each $i$ from 1 to $n$
   
   let $j = \left\lfloor \frac{a_i \cdot n}{r} \right\rfloor$ and add $a_i$ to $B_j$.  
3. Sort all buckets using Insertion Sort.
4. Output the concatenation of $B_1, \ldots, B_n$

- Steps 1,2, and 4 can be implemented to run in $O(n)$ time.

**Lemma**

Step 3 runs in expected time $O(n)$.  

4/26/2022

Sofya Raskhodnikova; Randomness in Computing
Bucket Sort: Analysis

**Lemma**

Step 3 (sorting the buckets) runs in expected time $O(n)$.

**Proof:** Buckets are bins, elements are balls in Balls-in-the-Bins.

- Let RV $X_j = \#$ of elements that land in bucket $B_j$, for $j = 1, \ldots, n$
- Time to sort $B_j$ is: $\leq c \cdot X_j^2$ for some constant $c$
- Expected run time of Step 3:

$$
\leq \mathbb{E} \left[ \sum_{j \in [n]} cX_j^2 \right] = c \cdot \sum_{j \in [n]} \mathbb{E}[X_j^2] = cn \cdot \mathbb{E}[X_1^2]
$$

by linearity of expectation  
by symmetry
Bucket Sort: Analysis

**Lemma**

Step 3 (sorting the buckets) runs in expected time $O(n)$.

**Proof**: Buckets are bins, elements are balls in Balls-in-the-Bins.

- Let RV $X_j = \#$ of elements that land in bucket $B_j$, for $j = 1, \ldots, n$
- Expected run time of Step 3: $\leq cn \cdot \mathbb{E}[X_1^2]$
- $X_1 \sim$
Bucket Sort: Conclusion

- **Given:** $n$ integers from range $[r]$.

**Theorem**

If $n$ integers are chosen uniformly and independently from range $\{1, \ldots, r\}$, they can be sorted in expected time $O(n)$. 
Probability is used in the design and analysis of algorithms

1. to analyze deterministic algorithms when inputs come from a specified distribution

2. to design and analyze algorithms that use randomness