Randomness in Computing

LECTURE 7

Last time
• Branching process
• Geometric RVs
• Coupon collector problem

Today
• Randomized quicksort
• Markov’s inequality
• Variance
Quicksort: divide and conquer

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

\[ A: \]

\[ \leq x \quad x \quad \geq x \]

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

\[ \leq x \quad x \quad \geq x \]

Sofya Raskhodnikova; based on notes by E. Demaine and C. Leiserson
QuickSort

QuickSort(array A, positive integers \( \ell, r \))

1. if \( \ell < r \)
2. then \( p \leftarrow \text{Partition}(A, \ell, r) \)
3. QuickSort \( (A, \ell, p - 1) \)
4. QuickSort \( (A, p + 1, r) \)

Initial call: QuickSort \( (A, 1, n) \)

Sofya Raskhodnikova; based on notes by E. Demaine and C. Leiserson
**Example of partitioning**

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>10</td>
<td>13</td>
<td>5</td>
<td>8</td>
<td>3</td>
<td>2</td>
<td>11</td>
</tr>
</tbody>
</table>

\[ i \quad j \]

*Sofya Raskhodnikova; based on notes by E. Demaine and C. Leiserson*
Example of partitioning

\[ 6 \quad 10 \quad 13 \quad 5 \quad 8 \quad 3 \quad 2 \quad 11 \]

\[ i \rightarrow j \]
Example of partitioning

Sofya Raskhodnikova; based on notes by E. Demaine and C. Leiserson
Example of partitioning

Sofya Raskhodnikova; based on notes by E. Demaine and C. Leiserson
Example of partitioning

<table>
<thead>
<tr>
<th>6</th>
<th>10</th>
<th>13</th>
<th>5</th>
<th>8</th>
<th>3</th>
<th>2</th>
<th>11</th>
</tr>
</thead>
</table>

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

$i$  $j$
Example of partitioning

Sofya Raskhodnikova; based on notes by E. Demaine and C. Leiserson
Example of partitioning

Sofya Raskhodnikova; based on notes by E. Demaine and C. Leiserson
Example of partitioning

Sofya Raskhodnikova; based on notes by E. Demaine and C. Leiserson
Example of partitioning

Sofya Raskhodnikova; based on notes by E. Demaine and C. Leiserson
Example of partitioning

Sofya Raskhodnikova; based on notes by E. Demaine and C. Leiserson
Example of partitioning

Sofya Raskhodnikova; based on notes by E. Demaine and C. Leiserson
Example of partitioning

<table>
<thead>
<tr>
<th>6</th>
<th>10</th>
<th>13</th>
<th>5</th>
<th>8</th>
<th>3</th>
<th>2</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>5</td>
<td>13</td>
<td>10</td>
<td>8</td>
<td>3</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>3</td>
<td>10</td>
<td>8</td>
<td>13</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>13</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>3</td>
<td>6</td>
<td>8</td>
<td>13</td>
<td>10</td>
<td>11</td>
</tr>
</tbody>
</table>

Sofya Raskhodnikova; based on notes by E. Demaine and C. Leiserson
### Partitioning algorithm

**Partition (array A, positive integers \( \ell, r \))**

1. \( x \leftarrow A[\ell] \)  // \( A[\ell] \) becomes the pivot
2. \( i \leftarrow \ell \)
3. **for** \( j = \ell + 1 \) **to** \( r \)
4. \hspace{1em} **if** \( A[j] < x \)
5. \hspace{2em} **then** \( i \leftarrow i + 1 \)
6. \hspace{2em} SWAP(A[i], A[j])
7. SWAP(A[\ell], A[i])
8. **return** \( i \)

---

**Sofya Raskhodnikova; based on notes by E. Demaine and C. Leiserson**
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.
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.

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

9/23/2021
Theorem. The expected number of comparisons is $2n \ln n + O(n)$.

Proof (continued):

- 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.
- $\mathbb{E}[X_{ij}] = \Pr[X_{ij} = 1]$
- **Important idea**: $y_i$ and $y_j$ are compared iff either $y_i$ or $y_j$ is the first pivot chosen from $Y_{ij} = \{y_i, \ldots, y_j\}$
- The first time a pivot is chose from $Y_{ij}$, it is equally likely to be any of $j - i + 1$ elements of $Y_{ij}$.
Theorem. The expected number of comparisons is $2n \ln n + O(n)$.

Proof (continued):

$$
\Pr[X_{ij} = 1] = \Pr[x_i \text{ or } x_j \text{ is the first pivot chosen from } Y_{ij}] = \frac{2}{j-i+1}
$$

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

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

**Theorem.** Let $X$ be a RV taking only nonnegative values. Then, for all $a > 0$,

$$\Pr[X \geq a] \leq \frac{\mathbb{E}[X]}{a}.$$ 

**Proof:** Let $a > 0$. 

9/23/2021
Markov’s Inequality

**Theorem.** Let $X$ be a RV taking only nonnegative values. Then, for all $a > 0$,

$$\Pr[X \geq a] \leq \frac{\mathbb{E}[X]}{a}.$$ 

**Alternative proof:** Let $a > 0$. 

Markov’s Inequality

- **Theorem.** Let \( X \) be a RV taking only nonnegative values. Then, for all \( a > 0 \),

\[
\Pr[X \geq a] \leq \frac{\mathbb{E}[X]}{a}.
\]

- **Alternatively:** Then, for all \( b > 1 \),

\[
\Pr[X \geq b \cdot \mathbb{E}[X]] \leq \frac{1}{b}.
\]

- **Example:** Show that Randomized Quicksort uses

\[
\leq 10n \ln n + O(n)
\]

comparisons with probability at least \( 4/5 \).
Bernoulli random walk

- Start at position 0.
- At every step go up or down by 1 with probability 1/2 each.
- Let $X$ be the position after $n$ steps.

What is the probability space?

A. Uniform over $\{0,1\}$.
B. Uniform over $\{-1,1\}$.
C. $2^n$.
D. Position after $n$ steps.
E. Uniform over $\{(s_1, \ldots, s_n) \mid s_i \in \{-1,1\} \text{ for } i = 1, \ldots, n\}$. 
Bernoulli random walk

• Start at position 0.
• At every step go up or down by 1 with probability 1/2 each.
• Let $X$ be the position after $n$ steps.
• What is $\mathbb{E}[X]$?
• How far from the origin should we expect $X$ to be?

A precise answer to this question is the expectation of $|X|$. However, it is easier to work with the expectation of $X^2$. (It is not the same! But gives us an idea.)
Random variables: variance

- The **variance** of a random variable $X$ with expectation $\mathbb{E}[X] = \mu$ is
  \[ \text{Var}[X] = \mathbb{E}[(X - \mu)^2]. \]

- Equivalently, $\text{Var}[X] = \mathbb{E}[X^2] - \mu^2$.

- The **standard deviation** of $X$ is $\sigma[X] = \sqrt{\text{Var}[X]}$. 