Homework

Divide and Conquer Exercise 3 due September 21

• Programming assignment (Java or Python)
• Closest pair of points
Randomness is a powerful tool for algorithm design:

- Avoiding worst-case performance
  - Balanced hash tables
  - Quicksort algorithm
  - Linear-time selection

- Trading off running time for small error probability
  - Polynomial identity testing
  - Primality testing

“Las Vegas” algorithms: Always succeeds and average running time is bounded (but worst-case running time may not be)

“Monte Carlo” algorithms: Algorithm may produce wrong answer (with small probability)
The Online Hiring Problem

Candidate | Score
--- | ---
[Image of candidates and scores]

**Hiring problem:** hire the most qualified candidate for a job
- Total of $n$ candidates (arrive in random order)
- We can associate a (unique) score with each candidate after an interview
- **Constraint:** we must make the decision to accept or decline a candidate immediately after the interview; we can hire exactly one candidate

How do we hire the most qualified candidate (with good probability)?
The Online Hiring Problem

**Candidate**

Score

2  4  8  3  9  6  7  5

**Hiring problem:** hire the most qualified candidate for a job

- **Total of** $n$ **candidates (arrive in random order)** sample space
- We can associate a (unique) score with each candidate after an interview
- **Constraint:** we must make the decision to accept or decline a candidate **immediately** after the interview; we can hire exactly one candidate

**Strategy 1:** Hire the first candidate

**Strategy 2:** Interview the first $k$ candidates, hire the first candidate in remaining $n - k$ whose score is higher than that of the first $k$ candidates
The Online Hiring Problem

Event $S$: success (i.e., we hire the most qualified candidate)
Event $S_i$: success and the most qualified candidate is the $i^{th}$ candidate

\[
\Pr[S] = \sum_{i=1}^{n} \Pr[S_i] \quad \forall i \leq k: \ Pr[S_i] = 0
\]

**Goal:** Choose $k$ to maximize the probability of hiring the most qualified candidate.
The Online Hiring Problem

Event $S_i$: success and the most qualified candidate is the $i^{th}$ candidate

Requirements:
- Most qualified candidate is the $i^{th}$ candidate
- Most qualified candidate among the first $i - 1$ candidates is in the first $k$

$$\forall i > k: \Pr[S_i] = \frac{1}{n} \cdot \frac{k}{i - 1}$$
The Online Hiring Problem

Event $S_i$: success and the most qualified candidate is the $i^{th}$ candidate

$$
\Pr[S] = \sum_{i=1}^{n} \Pr[S_i] = \sum_{i=k+1}^{n} \frac{k}{n(i-1)} = \frac{k}{n} \sum_{i=k}^{n-1} \frac{1}{i}
$$
The Online Hiring Problem

\[
\Pr[S] = \sum_{i=1}^{n} \Pr[S_i] = \sum_{i=k+1}^{n} \frac{k}{n(i-1)} = \frac{k}{n} \sum_{i=k}^{n-1} \frac{1}{i}
\]
Another Approach to the Hiring Problem

<table>
<thead>
<tr>
<th>Candidate</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>

**Hiring algorithm:**

- Interview each candidate in the order they arrive
- Hire the candidate if they are more qualified than the current candidate

What is the expected number of candidates we end up hiring?
Another Approach to the Hiring Problem

Let $X$ be the total number of candidates hired
Let $X_i = 1$ if we hire the $i^{th}$ candidate and $X_i = 0$ otherwise

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