# Chapter 5: Probabilistic Analysis and Randomised Algorithms

(Version of 8th January 2024)

Pierre Flener

Department of Information Technology
Computing Science Division
Uppsala University
Sweden

Course 1DL481:
Algorithms and Data Structures 3 (AD3)

# Hiring Problem

Suppose your company needs to hire a new employee:

- An employment agency provides a list of $n$ applicants.
- You interview one applicant per day.
- After each interview, you must immediately decide if you hire the interviewed applicant or not.
- You can rehire if a better applicant is interviewed later, but you must pay a fee to the fired employee.

How to hire the best applicant at minimal financial cost?
(Not at minimal runtime, as you interview all $n$ applicants.)

# Hiring Problem: Deterministic Algorithm

**Step 1.** Always hire after a better applicant was interviewed.

**Example:** For $n = 8$ applicants with post-interview scores $[2, 3, 1, 4, 6, 5, 8, 7]$, you hire 5 times (and pay the firing fee $5 - 1 = 4$ times).

Given $n$ applicants, how many times do you hire?

- Worst case: $n$ hirings.
  **Example:** For $n = 8$ applicants with post-interview scores $[1, 2, 3, 4, 5, 6, 7, 9]$, you hire $n = 8$ times.

- Average case: ?
  We need to perform a probabilistic analysis: we use **knowledge** of the distribution of the inputs, or we make **assumptions** about it.

For any particular input, the output and its cost are the same at each run.

# Hiring Problem: Probabilistic Analysis

UPPSALA UNIVERSITET

Define indicator random variables (independent ones here):

$$X_i = \begin{cases} 1 & \text{if applicant } i \text{ is hired} \\ 0 & \text{otherwise} \end{cases}$$

Let $E[X_i]$ denote the expected value of $X_i$, that is the probability that applicant $i$ is hired, that is the probability that applicant $i$ is better than 1 to $i - 1$, that is $1/i$ upon **assuming** that the applicants are randomly ordered.
Average number of hirings, much lower than the worst-case number:

$$\begin{aligned} E\left[\sum_{i=1}^{n} X_i\right] &= \sum_{i=1}^{n} E[X_i] && \text{by linearity of expectation} \\ &= \sum_{i=1}^{n} 1/i && \textbf{assuming} \text{ a random order} \\ &= \ln n + O(1) && \text{the } n\text{th harmonic number } H(n) \end{aligned}$$

UPPSALA UNIVERSITET

# Hiring Problem: Trust Issue

- We **assumed** that the agency's list is randomly ordered!

- But what if we cannot trust the employment agency?

**Example:** What if we must also pay the employment agency a fee each time we fire and rehire?

# Hiring Problem: Randomised Algorithm

The probabilistic analysis guides us to a new algorithm:

**Step 0.** Shuffle the applicants randomly before interviewing.
**Step 1.** Always hire after a better applicant was interviewed.

A randomised algorithm **imposes** a particular distribution, namely a random order, independently of the input order.

- Expected case: $\ln n + O(1)$ hirings, like the average case of the deterministic algorithm upon **assuming** a random order.
- Worst case: no particular input triggers $n$ hirings at every run!

For any particular input, the output and its cost can differ between runs.

UPPSALA UNIVERSITET

# Hiring Problem: Questions

- How to shuffle (permute) an array $A[1\,..\,n]$ randomly?

  Swap $A[i]$ with $A[\textsc{Random}(i,n)]$, for $i = 1$ to $n$,
  where $\textsc{Random}(\ell, u)$ returns in $\Theta(1)$ time a random integer between integers $\ell$ and $u$, inclusive.
  This takes $\Theta(n)$ time overall.

- On-line Hiring Problem: You can only hire one applicant (and you then stop interviewing the remaining ones).
  How do you hire a candidate close to the best?

  Reject the first $n/e$ applicants and hire the first one thereafter that scores higher than all preceding ones, or hire the last one (in case the best one was among the first $n/e$ ones). See Section 5.4.4 of CLRS4 for details.