# Probabilistic Analysis

August 5, 2014

## 1 Introduction

Once you realized the advantages of using probability for complexity analysis, it is clear that you require to have an average input to obtain the complexities given by the probabilistic analysis. Therefore, it is quite clear that the we need to be sure that the input is most of the time an average input. To exemplify this, we can look at the hiring problem.

## 2 The Hiring Problem

While looking at the hiring problem and the algorithm, we recognize the following: The job position is never empty. Then,

- We need to interview all the candidates.
- We hire $m$ of them.

Then, we get the following cost for the hiring problem in asymptotic notation: $O(nc_i + mc_h)$. Therefore, when looking a the worst case scenario, hiring all the candidates, we have the following complexity $O(nc_i + nc_h) = O(nc_h)$.

## 3 The Indicator Function

Here, we have an important lemma about the indicator function that will be used again and again on many of the probabilistic analysis.

**Lemma** Given a sample space $S$ and an event $A$ in the sample space S, let $X_A = I\{A\}$. Then, $E(X_A) = Pr\{A\}$.

Proof

Simply look at this $E(X_A) = E(I\{A\}) = 1 \times Pr\{A\} + 0 \times Pr\left\{\overline{A}\right\} = Pr\{A\}$.

For example, given a coin being flipped $n$ time $X_i = I\{\text{The } i\text{th flip result}\}$. Then, then we can count all the possible flips by using the following random variable:

$$X = \sum_{i=1}^{n} X_i \tag{1}$$

Therefore, taking the expected value in both sides, we have that

$$E[X] = E\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} E[X_i] = \sum_{i=1}^{n} \frac{1}{2} = \frac{n}{2} \tag{2}$$

# 4 Analyzing Random Hiring

Although the differences between normal hiring and randomized hiring could look extreme. Actually the proof of the complexity $O(c_h \log n)$ is the same once we notice that the permutation of the elements give us the same situation as in the hiring when the input order is drawn from a uniform distribution. Then, enforcing uniformity is the key piece of the entire idea of getting the expected complexity.

# 5 Enforcing Uniformity

We can see in the slides two different algorithms to enforce uniformity. Each of them uses a lemma to prove the enforced uniformity.

**Lemma** 5.4

Procedure Permute-by-sorting produces a uniform random permutation of the input, assuming that all probabilities are distinct.

**Proof:**

**Case** I

Assume $A[i]$ receives the ith smallest priority. Let $E_i$be the event that element $A[i]$ receives the ith smallest probability. Therefore, we wish to compute the event the probability of event $E_1 \cap E_2 \cap ... \cap E_n$, and given that we have$n!$ possible permutations of the ranking, we want to be sure that the $P(E_1 \cap E_2 \cap ... \cap E_n) = \frac{1}{n!}$. This event is the most naive one, it means that $A[1]$is the 1st element, $A[2]$ is the 2nd element and so on. Then, it is simply a case of using the chain rule:

$$\begin{gathered} P(E_1, E_2, ..., E_n) = P(E_n|E_1, ..., E_{n-1})P(E_{n-1}|E_1, ..., E_{n-2})... \\ P(E_2|E_1)P(E_1) \end{gathered} \tag{3}$$

First imagine that you have $n$ possible positions at your array and you need to fill them. Now look at this:

- At $E_1$ you have $n$ different elements to put at position one or $P(E_1) = \frac{1}{n}$.
- At $E_2$ you have $n-1$ different elements to put at position two or $P(E_2|E_1) = \frac{1}{n-1}$.
- etc.

  Therefore, $P(E_1, E, ..., E_n) = 1 \times \frac{1}{2} \times \frac{1}{3} \times ... \times \frac{1}{n} = \frac{1}{n!}$.

**Case** II

In the general case, we have that we can use for any permutation $\sigma = \langle \sigma(1), \sigma(2), ..., \sigma(n) \rangle$ of the set $\{1, 2, ..., n\}$. Let us to assign rank $r_i$ to the element $A[i]$. Then, if we define $E_i$ as the event in which element $A[i]$ receives the $\sigma(i)$th smallest priority $r_i = \sigma(i)$, we have the same proof.

**QED**

**Lemma** 5.5

Procedure RANDOMIZE-IN-PLACE computes a uniform random permutation.

**Proof:**

We use the following loop invariant, before entering lines 2-3 the array $A[1, ..., i-1]$ contains a $(i-1)$-permutation with probability $\frac{(n-i+1)!}{n!}$.

**Initialization.** We have an empty array $A[1, ..., 0]$ and $i = 1$, then $P(A[1, ..., 0]) = \frac{(n-i+1)!}{n!} = \frac{n!}{n!} = 1$. This is because of vacuity.

**Maintenance.** Then, by induction, we have that the array $A[1, ..., i-1]$ contains $(i-1)$-permutation with probability $\frac{(n-i+1)!}{n!}$. Now, consider the $i$-permutation contain the elements $\langle x_1, x_2, ..., x_i \rangle = \langle x_1, x_2, .., x_{i-1} \rangle \circ x_i$. Then, $E_1$denotes the event for the $(i-1)$-permutation with $P(E_1) = \frac{(n-i+1)!}{n!}$, and $E_2$ denotes putting element $x_i$ at position $A[i]$. Therefore, we have that $P(E_2 \cap E_1) = P(E_2|E_1)P(E_1) = \frac{1}{n-i+1} \times \frac{(n-i+1)!}{n!} = \frac{(n-i)!}{n!}$.

**Termination.** Now with $i = n + 1$ we have that the array $A[1, ..., n]$ contains a $n-$permutation with probability $\frac{(n-n+1)!}{n!} = \frac{1}{n!}$.