CHAPTER 3

Moments and Deviations

In Chapters 1 and 2, we bounded the expected running times of several randomized algorithms. While the expectation of a random variable (such as a running time) may be small, it may frequently assume values that are far higher. In analyzing the performance of a randomized algorithm, we often like to show that the behavior of the algorithm is good almost all the time. For example, it is more desirable to show that the running time is small with high probability, not just that it has a small expectation. In this chapter we will begin the study of general methods for proving statements of this type. We will begin by examining a family of stochastic processes that is fundamental to the analysis of many randomized algorithms: these are called *occupancy problems*. This motivates the study (in this chapter and the next) of general bounds on the probability that a random variable deviates far from its expectation, enabling us to avoid such custom-made analyses. The probability that a random variable deviates by a given amount from its expectation is referred to as a *tail probability* for that deviation. Readers wishing to review basic material on probability and distributions may consult Appendix C.

3.1. Occupancy Problems

We begin with an example of an *occupancy problem*. In such problems we envision each of m indistinguishable objects ("balls") being randomly assigned to one of n distinct classes ("bins"). In other words, each ball is placed in a bin chosen independently and uniformly at random. We are interested in questions such as: what is the maximum number of balls in any bin? what is the expected number of bins with k balls in them? Such problems are at the core of the analyses of many randomized algorithms ranging from data structures to routing in parallel computers. Later, in Section 3.6, we will encounter a variant of the occupancy problem, known as the *coupon collector's problem*; in

Chapter 4, we will apply sophisticated techniques to various random variables arising in occupancy problems.

Our discussion of the occupancy problem will illustrate a recurrent tool in the analysis of randomized algorithms: that *the probability of the union of events is no more than the sum of their probabilities.* This is a special case of the Boole-Bonferroni Inequalities (Proposition C.2) and can be formally stated as follows: for arbitrary events $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_n$, not necessarily independent,

$$\mathbf{Pr}[\cup_{i=1}^{n} \mathcal{E}_i] \leq \sum_{i=1}^{n} \mathbf{Pr}[\mathcal{E}_i].$$

This principle is extremely useful because it assumes nothing about the dependencies between the events. Thus, it enables us to analyze phenomena involving events with very complicated interactions, without having to unravel the interactions.

Consider first the case $m = n$. For $1 \leq i \leq n$, let X_i be the number of balls in the ith bin. Following Example 1.1, we have $\mathbf{E}[X_i] = 1$ for all i. Yet we do not expect that during a typical experiment every bin receives exactly one ball. Rather, we expect some bins to have no balls at all, and others to have many more than one.

Let us try now to make a statement of the form "with very high probability, no bin receives more than k balls," for a suitably chosen k. Let $\mathcal{E}_j(k)$ denote the event that bin j has k or more balls in it. We concentrate on analyzing $\mathcal{E}_1(k)$. The probability that bin 1 receives exactly i balls is

$$\binom{n}{i}\left(\frac{1}{n}\right)^i\left(1-\frac{1}{n}\right)^{n-i} \leq \binom{n}{i}\left(\frac{1}{n}\right)^i \leq \left(\frac{ne}{i}\right)^i\left(\frac{1}{n}\right)^i = \left(\frac{e}{i}\right)^i.$$

The second inequality results from an upper bound for binomial coefficients (Proposition B.2). Thus,

$$\mathbf{Pr}[\mathcal{E}_1(k)] \leq \sum_{i=k}^{n}\left(\frac{e}{i}\right)^i \leq \left(\frac{e}{k}\right)^k\left(1+\frac{e}{k}+\left(\frac{e}{k}\right)^2+\cdots\right). \tag{3.1}$$

Let $k^* = \lceil (3\ln n)/\ln\ln n \rceil$. Then,

$$\mathbf{Pr}[\mathcal{E}_1(k^*)] \leq \left(\frac{e}{k^*}\right)^{k^*}\frac{1}{1-e/k^*} \leq n^{-2}.$$

The same computation tells us that this upper bound applies to $\mathbf{Pr}[\mathcal{E}_i(k^*)]$ for all i, but can we say that *no bin* is likely to have more than k^* balls in it? For this we invoke the principle mentioned at the beginning of this section: the probability of the union of the events $\mathcal{E}_i(k^*)$ is no more than their sum. We obtain that

$$\mathbf{Pr}[\cup_{i=1}^{n} \mathcal{E}_i(k^*)] \leq \sum_{i=1}^{n} \mathbf{Pr}[\mathcal{E}_i(k^*)] \leq \frac{1}{n}.$$

Thus we have established:

Theorem 3.1: *With probability at least* $1 - 1/n$, *no bin has more than* $k^* = (3 \ln n)/\ln \ln n$ *balls in it.*

Interestingly, when m is of the order of $n \log n$, the bin with the most balls has about the same number of balls as the expected number of balls in any bin. This phenomenon is exploited in a number of randomized algorithms (see, for instance, Section 4.2).

Exercise 3.1: For $m = n \log n$, show that with probability $1 - o(1)$ every bin contains $O(\log n)$ balls.

We turn to a classic combinatorial problem. Suppose that m balls are randomly assigned to n bins. We study the probability of the event that they all land in distinct bins. The special case $n = 365$ is popular in mathematical lore as the *birthday problem.* The interpretation is that the 365 days of the year correspond to 365 bins, and the birthday of each of m people is chosen independently and uniformly from all 365 days (ignoring leap years). How large must m be before two people in the group are likely to share their birthdays?

Consider the assignment of the balls to the bins as a sequential process: we throw the first ball into a random bin, then the second ball, and so on. For $2 \leq i \leq m$, let $\mathcal{E}_i$ denote the event that the ith ball lands in a bin not containing any of the first $i - 1$ balls. We will bound $\mathbf{Pr}[\cap_{i=2}^{m} \mathcal{E}_i]$ from above. From (1.6), we can write

$$\mathbf{Pr}[\cap_{i=2}^{m} \mathcal{E}_i] = \mathbf{Pr}[\mathcal{E}_2]\mathbf{Pr}[\mathcal{E}_3 \mid \mathcal{E}_2]\mathbf{Pr}[\mathcal{E}_4 \mid \mathcal{E}_2 \cap \mathcal{E}_3] \cdots \mathbf{Pr}[\mathcal{E}_m \mid \cap_{i=2}^{m-1} \mathcal{E}_i].$$

Now, it is easy to compute $\mathbf{Pr}[\mathcal{E}_i \mid \cap_{j=2}^{i-1} \mathcal{E}_j]$: this is simply the probability that the ith ball lands in an empty bin given that the first $i - 1$ all fell into distinct bins, and is thus $1 - (i - l)/n$. Making use of the fact that $1 - x \leq e^{-x}$, we have

$$\mathbf{Pr}[\cap_{i=2}^{m} \mathcal{E}_i] \leq \prod_{i=2}^{m} \left(1 - \frac{i-1}{n}\right) \leq \prod_{i=2}^{m} e^{-(i-1)/n} = e^{-m(m-1)/2n}.$$

Thus, we see that for m equal to $\lceil \sqrt{2n} + 1 \rceil$, the probability that all m balls land in distinct bins is at most $1/e$; as m increases beyond this value, the probability drops rapidly.

3.2. The Markov and Chebyshev Inequalities

We have seen above that making statements about the probability that a random variable deviates far from its expectation may involve a detailed, problem-specific analysis. Often, one can avoid such detailed analyses by resorting to general inequalities on such tail probabilities.

We begin with the Markov inequality, a fundamental tool we will invoke repeatedly when we develop more sophisticated bounding techniques. Let X be a discrete random variable and $f(x)$ be any real-valued function. Then the expectation of $f(X)$ is given by (see Appendix C)

$$\mathbf{E}[f(X)] = \sum_x f(x)\mathbf{Pr}[X = x].$$

Theorem 3.2 (Markov Inequality): *Let Y be a random variable assuming only non-negative values. Then for all $t \in \mathbb{R}^+$,*

$$\mathbf{Pr}[Y \geq t] \leq \frac{\mathbf{E}[Y]}{t}.$$

Equivalently,

$$\mathbf{Pr}[Y \geq k\mathbf{E}[Y]] \leq \frac{1}{k}.$$

PROOF: Define a function $f(y)$ by $f(y) = 1$ if $y \geq t$, and 0 otherwise. Then $\mathbf{Pr}[Y \geq t] = \mathbf{E}[f(Y)]$. Since $f(y) \leq y/t$ for all y,

$$\mathbf{E}[f(Y)] \leq \mathbf{E}\left[\frac{Y}{t}\right] = \frac{\mathbf{E}[Y]}{t},$$

and the theorem follows. □

This is the tightest possible bound when we know only that Y is non-negative and has a given expectation. Unfortunately, the Markov inequality by itself is often too weak to yield useful results. The following exercise may help the reader appreciate this; it shows that the Markov inequality is tight only for rather uninteresting distributions.

Exercise 3.2: Given a positive integer k, describe a random variable X assuming only non-negative values, such that

$$\mathbf{Pr}[X \geq k\mathbf{E}[X]] = \frac{1}{k}.$$

The following generalization of Markov's inequality underlies its usefulness in deriving stronger bounds.

Exercise 3.3: Let Y be any random variable and h any non-negative real function. Show that for all $t \in \mathbb{R}^+$,

$$\mathbf{Pr}[h(Y) \geq t] \leq \frac{\mathbf{E}[h(Y)]}{t}.$$

We now show that the Markov inequality can be used to derive better bounds on the tail probability by using more information about the distribution of the random variable. The first of these is the Chebyshev bound, which is based on the knowledge of the variance of the distribution; we will apply this to the analysis of a simple randomized selection algorithm.

For a random variable X with expectation μ_X, its *variance* σ_X^2 is defined to be $\mathbf{E}[(X - \mu_X)^2]$. The *standard deviation* of X, denoted σ_X, is the positive square root of σ_X^2. (See Appendix C.)

Theorem 3.3 (Chebyshev's Inequality): *Let X be a random variable with expectation μ_X and standard deviation σ_X. Then for any $t \in \mathbb{R}^+$,*

$$\mathbf{Pr}[|X - \mu_X| \geq t\sigma_X] \leq \frac{1}{t^2}.$$

PROOF: First, note that

$$\mathbf{Pr}[|X - \mu_X| \geq t\sigma_X] = \mathbf{Pr}[(X - \mu_X)^2 \geq t^2\sigma_X^2].$$

The random variable $Y = (X - \mu_X)^2$ has expectation σ_X^2, and applying the Markov inequality to Y bounds this probability from above by $1/t^2$. □

3.3. Randomized Selection

We now consider the use of random sampling for the problem of selecting the kth smallest element in a set S of n elements drawn from a totally ordered universe. We assume that the elements of S are all distinct, although it is not very hard to modify the following analysis to allow for multisets. Let $r_S(t)$ denote the rank of an element t (the kth smallest element has rank k) and let $S_{(i)}$ denote the ith smallest element of S. We extend the use of this notation to subsets of S as well. Thus we seek to identify $S_{(k)}$.

In Step 1 (see following page), we sample with replacement: for instance, if an element s of S is chosen to be in R on the first of our $n^{3/4}$ drawings, the remaining $n^{3/4} - 1$ drawings are all as likely to pick s again as any other element in S. This style of sampling appears to be wasteful, but we employ it here because it keeps our analysis clean. Sampling without replacement would result in a marginally sharper analysis, but in practice this may be slightly harder to implement: throughout the sampling process, we would have to keep track of the elements chosen so far.

Figure 3.1 illustrates Step 3, where small elements are at the left end of the picture and large ones at the right. Determining (in Step 4) whether $S_{(k)} \in P$ is easy since we know the ranks $r_S(a)$ and $r_S(b)$ and we compare either or both of these to k, depending on which of the three **if** statements in Step 4 we execute. The sorting in Step 5 can be performed in $O(n^{3/4} \log n)$ steps.

Algorithm LazySelect:

Input: A set S of n elements from a totally ordered universe, and an integer k in $[1, n]$.

Output: The kth smallest element of S, $S_{(k)}$.

1. Pick $n^{3/4}$ elements from S, chosen independently and uniformly at random with replacement; call this multiset of elements R.
2. Sort R in $O(n^{3/4} \log n)$ steps using any optimal sorting algorithm.
3. Let $x = kn^{-1/4}$. For $\ell = \max\{\lfloor x - \sqrt{n} \rfloor, 1\}$ and $h = \min\{\lceil x + \sqrt{n} \rceil, n^{3/4}\}$, let $a = R_{(\ell)}$ and $b = R_{(h)}$. By comparing a and b to every element of S, determine $r_S(a)$ and $r_S(b)$.
4. **if** $k < n^{1/4}$, **then** $P = \{y \in S \mid y \leq b\}$;
   else if $k > n - n^{1/4}$, let $P = \{y \in S \mid y \geq a\}$;
   else if $k \in [n^{1/4}, n - n^{1/4}]$, let $P = \{y \in S \mid a \leq y \leq b\}$;
   Check whether $S_{(k)} \in P$ *and* $|P| \leq 4n^{3/4} + 2$. If not, repeat Steps 1–3 until such a set P is found.
5. By sorting P in $O(|P| \log |P|)$ steps, identify $P_{(k - r_S(a)+1)}$, which is $S_{(k)}$.

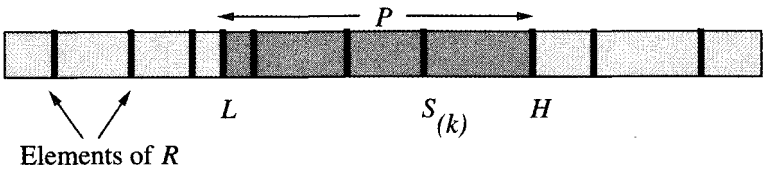

Figure 3.1: The **LazySelect** algorithm.

Thus the idea of the algorithm is to identify two elements a and b in S such that both of the following statements hold with high probability:

1. The element $S_{(k)}$ that we seek is in P.
2. The set P of elements between a and b is not very large, so that we can sort P inexpensively in Step 5.

We examine how either of these requirements could fail. We focus on the most interesting case when $k \in [n^{1/4}, n - n^{1/4}]$, so that $P = \{y \in S \mid a \leq y \leq b\}$; the analysis for the other two cases of Step 4 is similar and in fact somewhat simpler.

If the element a is greater than $S_{(k)}$ (or if b is smaller than $S_{(k)}$), we fail because P does not contain $S_{(k)}$. For this to happen, fewer than ℓ of the samples in R should be smaller than $S_{(k)}$ (respectively, at least h of the random samples should be smaller than $S_{(k)}$). We will bound the probability that this happens using the Chebyshev bound.

The second type of failure occurs when P is too big. To study this, we define $k_\ell = \max\{1, k - 2n^{3/4}\}$ and $k_h = \min\{k + 2n^{3/4}, n\}$. To obtain an upper bound on the probability of this kind of failure, we will be pessimistic and say that failure occurs if either $a < S_{(k_\ell)}$ or $b > S_{(k_h)}$. We prove that this is also unlikely, again using the Chebyshev bound. Before we perform this analysis, we establish an important property of independent random variables. Recall the definition of a joint density function $p(x, y)$ for random variables X and Y (Definition C.9).

► **Definition 3.1:** Let X and Y be random variables and $f(x, y)$ be a function of two real variables. Then,

$$\mathbf{E}[f(X, Y)] = \sum_{x,y} f(x, y)p(x, y).$$

For independent random variables X and Y we have from Proposition C.6

$$\mathbf{E}[XY] = \mathbf{E}[X]\mathbf{E}[Y]. \tag{3.2}$$

Lemma 3.4: *Let $X_1, X_2, \ldots, X_m$ be independent random variables. Let $X = \sum_{i=1}^m X_i$. Then $\sigma_X^2 = \sum_{i=1}^m \sigma_{X_i}^2$.*

PROOF: Let μ_i denote $\mathbf{E}[X_i]$, and $\mu = \sum_{i=1}^m \mu_i$. The variance of X is given by

$$\mathbf{E}[(X - \mu)^2] = \mathbf{E}[(\sum_{i=1}^m (X_i - \mu_i))^2].$$

Expanding the latter and using linearity of expectations, we obtain

$$\mathbf{E}[(X - \mu)^2] = \sum_{i=1}^m \mathbf{E}[(X_i - \mu_i)^2] + 2\sum_{i<j} \mathbf{E}[(X_i - \mu_i)(X_j - \mu_j)].$$

Since all pairs X_i, X_j are independent, so are the pairs $(X_i - \mu_i), (X_j - \mu_j)$. By (3.2), each term in the latter summation can be replaced by $\mathbf{E}[(X_i - \mu_i)]$ $\mathbf{E}[(X_j - \mu_j)]$. Since $\mathbf{E}[(X_i - \mu_i)] = \mathbf{E}[X_i] - \mu_i = 0$, the latter summation vanishes. It follows that

$$\mathbf{E}[(X - \mu)^2] = \sum_{i=1}^m \mathbf{E}[(X_i - \mu_i)^2] = \sum_{i=1}^m \sigma_{X_i}^2.$$

□

As in the analysis of **RandQS** in Chapter 1, we measure the running time of **LazySelect** in terms of the number of comparisons performed by it.

Theorem 3.5: *With probability $1 - O(n^{-1/4})$, **LazySelect** finds $S_{(k)}$ on the first pass through Steps 1–5, and thus performs only $2n + o(n)$ comparisons.*

PROOF: The time bound is easily established by examining the algorithm; Step 3 requires $2n$ comparisons, and all other steps perform $o(n)$ comparisons, provided the algorithm finds $S_{(k)}$ on the first pass through Steps 1–5. We now consider

the first mode of failure listed above: $a > S_{(k)}$ because fewer than ℓ of the samples in R are less than or equal to $S_{(k)}$ (so that $S_{(k)} \notin P$). Let $X_i = 1$ if the ith random sample is at most $S_{(k)}$, and 0 otherwise; thus $\mathbf{Pr}[X_i = 1] = k/n$, and $\mathbf{Pr}[X_i = 0] = 1 - k/n$. Let $X = \sum_{i=1}^{n^{3/4}} X_i$ be the number of samples of R that are at most $S_{(k)}$. Note that we really do mean the number of samples, and not the number of distinct elements. The random variables X_i are *Bernoulli trials* (Appendix C): each may be thought of as the outcome of a coin toss. Then, using Lemma 3.4 and the variance of a Bernoulli trial with success probability p

$$\mu_X = \frac{kn^{3/4}}{n} = kn^{-1/4},$$

and

$$\sigma_X^2 = n^{3/4}\left(\frac{k}{n}\right)\left(1 - \frac{k}{n}\right) \leq \frac{n^{3/4}}{4}.$$

This implies that $\sigma_X \leq n^{3/8}/2$. Applying the Chebyshev bound to X,

$$\mathbf{Pr}[|X - \mu_X| \geq \sqrt{n}] \leq \mathbf{Pr}[|X - \mu_X| \geq 2n^{1/8}\sigma_X] = \mathrm{O}\left(n^{-1/4}\right).$$

An essentially identical argument shows that

$$\mathbf{Pr}[b < S_{(k)}] = \mathrm{O}\left(n^{-1/4}\right).$$

Since the probability of the union of events is at most the sum of their probabilities, the probability that either of these events occurs (causing $S_{(k)}$ to lie outside P) is $\mathrm{O}(n^{-1/4})$.

Now for the second mode of failure – that P contains more than $4n^{3/4} + 2$ elements. For this, the analysis is very similar to that above in studying the first mode of failure, with k_ℓ and k_h playing the role of k. The analysis shows that $\mathbf{Pr}[a < S_{(k_\ell)}]$ and $\mathbf{Pr}[b > S_{(k_h)}]$ are both $\mathrm{O}(n^{-1/4})$ (the reader should verify these details). Adding up the probabilities of all of these failure modes, we find that the probability that Steps 1–3 fail to find a suitable set P is $\mathrm{O}(n^{-1/4})$. □

Exercise 3.4: The failure probability can be driven down further at the expense of increased running time. For a suitable definition of the $o(n)$ term, give an upper bound on the probability that the algorithm does not find $S_{(k)}$ in $cn + o(n)$ steps for $c > 2$.

Exercise 3.5: Theorem 3.5 tells us that the probability that **LazySelect** terminates in $2n + o(n)$ steps goes to 1 as $n \to \infty$. Suggest a modification in the algorithm that brings the constant in the linear term down to 1.5 from 2. We will refine this further in Problem 4.15.

This adds to the significance of **LazySelect**: the best known deterministic selection algorithms use $3n$ comparisons in the worst case and are quite complicated to implement. Further, it is known that any deterministic algorithm for

finding the median requires at least $2n$ comparisons, so we have a randomized algorithm that is both fast and has an expected number of comparisons that is provably smaller than that of any deterministic algorithm. The high probability bound of the previous exercise can be easily converted into a bound on the expected running time:

Exercise 3.6: Show that as a direct corollary of Theorem 3.5, the expected running time of the **LazySelect** algorithm is $2n + o(n)$.

Consider what happens when we modify **LazySelect** to be recursive as follows: in Step 5, instead of sorting P we recursively use **LazySelect** to find $P_{(k-rs(a)+1)}$. In this recursive version, the size of the candidate set P in which we are seeking $S_{(k)}$ is shrinking as the recursion proceeds. Using our analysis we can prove that at a typical stage of recursion the probability of failure at that stage is $O(|P|^{-1/4})$. But $|P|$ is diminishing, so that this probability of failure is rising as the algorithm proceeds! Thus, when the candidate set is down to a constant size, for instance, the failure probability is up to a constant and there is very little we can do about it. This is a fundamental barrier, not a weakness of our analysis. This is a typical problem with recursive randomized algorithms, and rears its head again in parallel randomized algorithms (where we always try to break a problem into smaller sub-problems) as well. A standard solution is to stop the recursion when the problem size is down to a certain size, and switch to a different, more expensive but deterministic technique – as we did by sorting in Step 5 of **LazySelect**.

3.4. Two-Point Sampling

We have so far been making use of the fact that the variance of the sum of *independent* random variables equals the sum of their variances. In fact, we can make a stronger statement. Let X and Y be discrete random variables defined on the same probability space. The *joint density function* of X and Y is the function

$$p(x, y) = \mathbf{Pr}[\{X = x\} \cap \{Y = y\}].$$

Thus $\mathbf{Pr}[Y = y] = \sum_x p(x, y)$, and

$$\mathbf{Pr}[X = x \mid Y = y] = \frac{p(x, y)}{\mathbf{Pr}[Y = y]}.$$

These definitions extend to a set $X_1, X_2, \ldots$ of more than two random variables. Such a set of random variables is said to be *pairwise independent* if for all $i \neq j$, and $x, y \in \mathbb{R}$,

$$\mathbf{Pr}[X_i = x \mid X_j = y] = \mathbf{Pr}[X_i = x].$$

We will use the result from the following exercise.

Exercise 3.7: Let n be a prime number and $\mathbb{Z}_n$ denote the ring of integers modulo n. For a and b chosen independently and uniformly at random from $\mathbb{Z}_n$, let $Y_i = ai + b \bmod n$. Show that for $i \not\equiv j \pmod n$, Y_i and Y_j are uniformly distributed on $\mathbb{Z}_n$ and pairwise independent. (Make use of the fact that in the field $\mathbb{Z}_n$, given fixed values for y_i and y_j, we can solve $y_i \equiv ai + b \pmod n$ and $y_j \equiv aj + b \pmod n$ uniquely for a and b.)

The following exercise is similar to Lemma 3.4.

Exercise 3.8: Let $X_1, X_2, \ldots, X_m$ be *pairwise* independent random variables, and $X = \sum_{i=1}^{m} X_i$. Show that $\sigma_X^2 = \sum_{i=1}^{m} \sigma_{X_i}^2$.

We now consider an application of these concepts to the reduction of the number of random bits used by ***RP*** algorithms (see Definition 1.8). Consider an ***RP*** algorithm A for deciding whether input strings x belong to a language L. Given x, A picks a random number r from the range $\mathbb{Z}_n = \{0, \ldots, n-1\}$, for a suitable choice of a prime n, and computes a binary value $A(x, r)$ with the following properties:

- If $x \in L$, then $A(x, r) = 1$ for at least half the possible values of r.
- If $x \notin L$, then $A(x, r) = 0$ for all possible choices of r.

For a randomly chosen r, $A(x, r) = 1$ is conclusive proof that $x \in L$, while $A(x, r) = 0$ is evidence that $x \notin L$.

For any $x \in L$, we refer to the values of r for which $A(x, r) = 1$ as *witnesses* for x; clearly, at least $n/2$ of the n possible values of r are witnesses. Of course, for $x \notin L$, there are no witnesses at all. The definition allows different $x \in L$ to have different sets of witnesses. Generally, n will be too large for us to test efficiently all the n potential witnesses for a given input x. However, for any $x \in L$, a random choice of r is a witness with probability at least $1/2$.

The fear is that $x \in L$ but the randomly chosen value of r yields $A(x, r) = 0$. However, we can drive down this probability of incorrectly classifying x by picking $t > 1$ values $r_1, \ldots, r_t$ independently from the range $\mathbb{Z}_n$, and computing $A(x, r_i)$ for all of them – in other words, by performing t independent iterations of the algorithm A on the same input x. If for any i we obtain $A(x, r_i) = 1$, we declare that x is in L, else we declare that x is not in L. By the independence of the trials, we are guaranteed that the probability of incorrectly classifying an input $x \in L$ (by declaring that it is not in L) is at most 2^{-t}.

Choosing t independent random numbers is expensive in that it requires $\Omega(t \log n)$ random bits. Suppose instead that we are only willing to use $O(\log n)$ random bits. In particular suppose that we wish to use only two independent samples from $\mathbb{Z}_n$. For a, b chosen independently from $\mathbb{Z}_n$, the naive usage of a and b as potential witnesses, i.e., computing $A(x, a)$ and $A(x, b)$, yields an upper

bound of only $1/4$ on the probability of incorrect classification. Here is a better scheme: let $r_i = ai + b \bmod n$, and compute $A(x, r_i)$ for $1 \leq i \leq t$. As before, if for any i we obtain $A(x, r_i) = 1$, we declare that x is in L, else we declare that x is not in L. What is the probability of incorrectly classifying any input x? We show that this probability is much smaller than $1/4$.

We need to worry about the possibility of making error only in the case where the input x is in L. Our analysis will be insensitive to the actual values of r in $\mathbb{Z}_n$ which are witnesses for x; we will only rely on the fact that at least half the values of r are witnesses. Clearly $A(x, r_i)$ is a random variable over the probability space of pairs a and b chosen independently from $\mathbb{Z}_n$. By the result of Exercise 3.7, the random r_i's are pairwise independent and, therefore, so are the random variables $A(x, r_i)$, for $1 \leq i \leq t$. Let $Y = \sum_{i=1}^{t} A(x, r_i)$. Assuming that $x \in L$, $\mathbf{E}[Y] \geq t/2$ and $\sigma_Y^2 \leq t/4$, or $\sigma_Y \leq \sqrt{t}/2$. The probability that the pairwise independent iterations produce an incorrect classification corresponds to the event $\{Y = 0\}$, and

$$\mathbf{Pr}[Y = 0] \leq \mathbf{Pr}[|Y - \mathbf{E}[Y]| \geq t/2].$$

By the Chebyshev inequality, the latter is at most $1/t$. Thus, the error probability is at most $1/t$, which is a considerable improvement over the error bound of $1/4$ achieved by the naive use of a and b. This improvement is sometimes referred to as *probability amplification.*

For a random variable X with expectation μ_X, we define the kth *central moment* to be $\mu_X^k = \mathbf{E}[(X - \mu_X)^k]$, if it exists (Appendix C). For example, the variance is the second central moment.

Exercise 3.9: The use of the variance of a random variable in bounding its deviation from its expectation is called *the second moment method.* In an analogous fashion, we can speak of the *kth moment method*: let k be even, and suppose we have a random variable X for which $\mu_X^k = \mathbf{E}[(X - \mu_X)^k]$ exists. Show that

$$\mathbf{Pr}[|X - \mu_X| > t\sqrt[k]{\mu_X^k}] \leq \frac{1}{t^k}.$$

Why is the kth moment method difficult to invoke for odd values of k?

The second moment method is generally useful for a random variable X if σ_X is $o(\mu_X)$. In a manner similar to "two-point" sampling (the name comes from the independent choice of two points a and b from which the r_i are derived), one can speak of k-point sampling for $k > 2$. The reader is referred to Appendix C for a further discussion of k-wise independence.

3.5. The Stable Marriage Problem

Consider a society in which there are n men (denoted by capital letters A,B,C, ...) and n women (denoted by a,b,c...). A *marriage* M is a 1-1 correspon-

dence between the men and the women. Assume a monogamous, heterosexual society. Each person has a preference list of the members of the opposite sex organized in a decreasing order of desirability. A marriage is said to be *unstable* if there exist two married couples X-x and Y-y such that X desires y more than x, and y desires X more than Y, implying that X-y will have a tendency to leave their current mates to marry each other. The pair X-y is said to be *dissatisfied* under this marriage. A marriage M in which there are no dissatisfied couples is called a *stable marriage.*

► **Example 3.1:**

For $n = 4$, consider the following preference lists.

$$\begin{array}{llll} A : abcd & B : bacd & C : adcb & D : dcab \\ a : ABCD & b : DCBA & c : ABCD & d : CDAB \end{array}$$

Consider the marriage M given by A-a, B-b, C-c, and D-d. Here C-d is a dissatisfied couple, implying that M is unstable. However, if C and d marry each other, and c and D marry each other, we obtain the stable marriage given by A-a, B-b, C-d, D-c.

The problem of finding stable marriages has several interesting applications, for example in matching medical graduates to residency positions in hospitals. It can be shown that for every choice of preference lists there exist at least one stable marriage. (Curiously enough, this is not the case in a homosexual, monogamous society with an even number of inhabitants.) We will prove this by presenting an algorithm to find a stable marriage. The naive approach of starting with an arbitrary marriage and trying to stabilize it by pairing up dissatisfied couples does not work.

Fortunately, an equally simple algorithm – the *Proposal Algorithm* – does the trick. The basic idea behind this algorithm can be summarized as "man proposes, woman disposes": each currently unattached man proposes to the most desirable woman on his list who has not already rejected him, and this woman then decides whether to accept or reject a proposal. The Proposal Algorithm is used by hospitals in North America in the match program that assigns medical graduates to residency positions.

More precisely, at any step, this algorithm will have a partial marriage. Assume that the men are numbered in some arbitrary manner. The lowest-numbered unmarried man X proposes to the most desirable woman on his list who has not already rejected him, call her x. The woman x will accept the proposal if she is currently unmarried, or if her current mate Y is less desirable to her than X (poor Y is jilted and reverts to the unmarried state). The algorithm repeats this process, terminating when every person has been married.

We show that this algorithm always terminates with a stable marriage. A woman once married will stay married during the course of the algorithm, although her mates may change with time. Furthermore, the desirability of her mates (in her view) can only improve with time. Thus at each step either a

woman gets married for the first time, or an already married woman obtains a more desirable mate.

An unattached man always has at least one woman available that he can proposition. This is because every woman he has already proposed to is currently married, and if he runs out of women then all women are married – this cannot happen unless all men are married too. Since at each step the proposer will eliminate one woman on his list, and the total size of the lists is n^2, we conclude that the algorithm uses at most n^2 proposals.

We claim that the final marriage M is stable. Otherwise, let X-y be a dissatisfied pair, where in M they are paired as X-x and Y-y. Since X prefers y to x, he must have proposed to y before getting married to x. Since y either rejected X, or accepted him only to jilt him later, her mates thereafter (including Y) must be more desirable to her than X. Therefore, y must prefer Y to X, contradicting the assumption that y is dissatisfied.

Our interest here is in performing an average-case analysis of this algorithm. Thus we are considering a probabilistic analysis of a deterministic algorithm. We introduce this analysis here because it touches upon several tools that are important in the analysis of randomized algorithms.

For this average-case analysis, we assume that the men's lists are chosen independently and uniformly at random; the women's lists can be arbitrary but must be fixed in advance. Let the random variable T_P denote the number of proposals made during the execution of the Proposal Algorithm. It is clear that the running time of the algorithm is proportional to T_P. At first glance, it may appear that the distribution T_P is extremely difficult to analyze, owing to the various dependencies between the proposals. For instance, the choice of the proposer at any step is severely conditioned by the history of the process. The choice of the woman at each step also depends on the past proposals of the current proposer.

We present a very simple technique – the *Principle of Deferred Decisions* – for getting around such problems using the example of the card game called *Clock Solitaire*. In this game we start with a standard deck of 52 cards, which is assumed to be randomly shuffled. The pack is then divided into 13 piles of 4 cards each. Each pile is arbitrarily labeled with a distinct member of $\{A, 2, 3, \ldots, J, Q, K\}$. On the first move we draw a card from the pile labeled K. At each subsequent move, a card is drawn from the pile whose label is the face value of the card drawn at the previous move (the suits of the cards are ignored in this game). The game ends when an attempt is made to draw a card from an empty pile. We win the game if, on termination, all 52 cards have been drawn; in all other cases we lose the game.

Let us estimate the probability of winning the game. Observe that the game always terminates in an attempt to draw a card from the K pile: the last card drawn has to be a K. This is because there are 4 cards of each denomination, and except for the K pile, each pile initially has 4 cards.

A naive view of the probability space for this game considers all possible ways of dealing out the cards. Each point in this space corresponds to some

partition of the 52 cards into 13 distinct piles, with an ordering defined on the 4 cards in each pile. Using this approach, computing the probability of a win would be a formidable task, since at each move of the game we introduce a new source of dependency.

We now examine a second probability space that better captures the dynamics of the game. The idea is to let the random choices unfold with the progress of the game, rather than fix the entire set of choices in advance. At each draw any unseen card is equally likely to appear. Thus, the process of playing this game is exactly equivalent to repeatedly drawing a card uniformly at random from a deck of 52 cards. A winning game corresponds to the situation where the first 51 cards drawn in this fashion contain exactly 3 Kings. The probability of the 52nd card drawn being a King is exactly 1/13; this is also the probability of winning the game.

The idea of the Principle of Deferred Decisions is to not assume that the entire set of random choices is made in advance. Rather, at each step of the process we fix only the random choices that must be revealed to the algorithm.

The Principle of Deferred Decisions can be used to simplify the average-case analysis of the Proposal Algorithm as follows. We do not assume that the men have chosen their (random) preference list in advance. In fact, let us suppose that men do not know their lists to start with. Each time a man has to make a proposal, he picks a random woman from the set of women not already propositioned by him, and proceeds to propose to her. Clearly, this is equivalent to choosing the random preference lists prior to the execution of the algorithm.

The only dependency that remains is that the random choice of a woman at any step depends on the set of proposals made so far by the current proposer. We can eliminate even this dependency, albeit at the cost of modifying the behavior of the algorithm. Suppose that each time a man makes a proposal, he chooses a woman uniformly at random from the set of *all* n women, including those to whom he has already proposed. In other words, he forgets the fact that these women have already rejected him. Call this new algorithm the Amnesiac Algorithm.

How does the performance of the new algorithm relate to that of the original one? Every proposal a man makes to a woman who has already rejected him will be rejected again. Thus, the output produced by the Amnesiac Algorithm is exactly the same as that of the original Proposal Algorithm. The only difference is that there are some wasted proposals in the Amnesiac Algorithm. Let T_A denote the number of proposals made by the Amnesiac Algorithm. Clearly, T_A *stochastically dominates* T_P (Appendix C): for all m, $\mathbf{Pr}[T_A > m] \geq \mathbf{Pr}[T_P > m]$. Therefore, it suffices for an upper bound to analyze the distribution of T_A.

A benefit of analyzing T_A is that we need only count the total number of proposals made, without regard to the name of the proposer at each stage. This is because each proposal is independently made to one of the n women chosen uniformly at random. Moreover, the algorithm terminates with a stable marriage once all women have received at least one proposal each. As will become clear shortly, bounding the value of T_A is a special case of the Coupon Collector's

Problem described in the next section. The following theorem is implied by Theorem 3.8, a result about deviations in the Coupon Collector's Problem that we will prove below in Section 3.6.

Theorem 3.6: *For any constant* $c \in \mathbb{R}$, *and* $m = n \ln n + cn$,

$$\lim_{n \to \infty} \mathbf{Pr}[T_A > m] = 1 - e^{-e^{-c}}.$$

3.6. The Coupon Collector's Problem

In the coupon collector's problem, there are n types of coupons and at each trial a coupon is chosen at random. Each random coupon is equally likely to be of any of the n types, and the random choices of the coupons are mutually independent. Let m be the number of trials. The goal is to study the relationship between m and the probability of having collected at least one copy of each of the n types. The reader may wish to make the correspondence between this process and an occupancy problem (Section 3.1) in which m balls are randomly distributed in n bins. This process will arise again in the study of random walks (Chapter 6). In this section we provide an amazingly precise answer to this question, while illustrating some fundamental ideas in the analysis of stochastic processes of the type that arise in randomized algorithms.

3.6.1. An Elementary Analysis

Let X be a random variable defined to be the number of trials required to collect at least one of each type of coupon. We first determine the expected value of X. Let $C_1, C_2, \ldots, C_X$ denote the sequence of trials, where $C_i \in \{1, \ldots, n\}$ denotes the type of the coupon drawn in the ith trial. Call the ith trial C_i a *success* if the type C_i was not drawn in any of the first $i-1$ selections. Clearly C_1 and C_X are always successes.

We divide the sequence into *epochs*, where epoch i begins with the trial following the ith success and ends with the trial on which we obtain the $(i+1)$st success. Define the random variable X_i, for $0 \leq i \leq n-1$, to be the number of trials in the ith epoch, so that

$$X = \sum_{i=0}^{n-1} X_i.$$

Further, let p_i denote the probability of success on any trial of the ith epoch. This is the probability of drawing one of the $n-i$ remaining coupon types and so,

$$p_i = \frac{n-i}{n}.$$

The random variable X_i is geometrically distributed with parameter p_i (see

Appendix C). Thus, the expected value of X_i is $1/p_i$ and its variance is $(1-p_i)/p_i^2$.

By linearity of expectation,

$$\mathbf{E}[X] = \mathbf{E}[\sum_{i=0}^{n-1} X_i] = \sum_{i=0}^{n-1} \mathbf{E}[X_i] = \sum_{i=0}^{n-1} \frac{n}{n-i} = n \sum_{i=1}^{n} \frac{1}{i} = nH_n.$$

By Proposition B.4 the nth Harmonic number H_n is asymptotically equal to $\ln n + \Theta(1)$, implying that

$$\mathbf{E}[X] = n \ln n + \mathrm{O}(n).$$

Since the X_i's are independent, we can determine the variance of X using Proposition C.9.

$$\begin{aligned} \sigma_X^2 &= \sum_{i=0}^{n-1} \sigma_{X_i}^2 \\ &= \sum_{i=0}^{n-1} \frac{ni}{(n-i)^2} \\ &= \sum_{i=1}^{n} \frac{n(n-i)}{i^2} \\ &= n^2 \sum_{i=1}^{n} \frac{1}{i^2} - nH_n. \end{aligned}$$

The sum $\sum_{i=1}^{n} 1/i^2$ converges to the constant $\pi^2/6$ for n approaching ∞; hence

$$\lim_{n\to\infty} \frac{\sigma_X^2}{n^2} = \frac{\pi^2}{6}.$$

Our next goal is to derive sharper estimates of the typical value of X. More precisely, we will show that the value of X is unlikely to deviate far from its expectation, or is *sharply concentrated around its expected value.* This entails bounding the tail probabilities of the distribution of X. The second moment method does not go far toward establishing such a result.

Exercise 3.10: Use the Chebyshev inequality to find an upper bound on the probability that $X > \beta n \ln n$, for a constant $\beta > 1$.

Let $\mathcal{E}_i^r$ denote the event that coupon type i is *not* collected in the first r trials. Using Proposition B.3 (Appendix B), we obtain that

$$\mathbf{Pr}[\mathcal{E}_i^r] = \left(1 - \frac{1}{n}\right)^r \leq e^{-r/n}.$$

This bound is $n^{-\beta}$ for $r = \beta n \ln n$.

Using the fact that the probability of a union of events is always less than the sum of the probabilities of these events, we obtain for $r = \beta n \ln n$,

$$\mathbf{Pr}[X > r] = \mathbf{Pr}[\cup_{i=1}^{n} \mathcal{E}_i^r] \leq \sum_{i=1}^{n} \mathbf{Pr}[\mathcal{E}_i^r] \leq \sum_{i=1}^{n} n^{-\beta} = n^{-(\beta-1)}.$$

We now study the probability that X deviates from its expectation nH_n by the amount cn, for any real-valued constant c. We will see that this probability drops very quickly as we increase the absolute value of c.

3.6.2. The Poisson Heuristic

Before we show the sharp concentration result for X, the following heuristic argument will help to establish some intuition. The heuristic argument is based on the approximation of the binomial distribution by the Poisson distribution (see Appendix C for definitions of these distributions). The material in this section, although useful, is not an essential prerequisite for subsequent topics and may be omitted in the first reading.

Let N_i^r denote the number of times the coupon of type i is chosen during the first r trials; the event $\mathcal{E}_i^r$ is the same as the event $\{N_i^r = 0\}$. The random variable N_i^r has the binomial distribution with parameters r and $p = 1/n$ (see Appendix C). This means that the probability that $N_i^r = x$, for $0 \leq x \leq r$, is as follows:

$$\mathbf{Pr}[N_i^r = x] = \binom{r}{x} p^x (1-p)^{r-x}.$$

Let λ be a positive real number. A (non-negative integer) random variable Y has the Poisson distribution with parameter λ if for any non-negative integer y,

$$\mathbf{Pr}[Y = y] = \frac{\lambda^y e^{-\lambda}}{y!}.$$

For suitably small λ and as r approaches ∞, the Poisson distribution with parameter $\lambda = rp$ is a good approximation to the binomial distribution with parameters r and p. In the current setting, we can approximate the distribution of N_i^r by the Poisson distribution with parameter $\lambda = r/n$. We will ignore the fact that λ may not be "suitably small" and that there could be significant error in this approximation; after all, this is only intended to be a heuristic calculation. Using this approximation, we calculate the probability of the event $\mathcal{E}_i^r$ as follows:

$$\mathbf{Pr}[\mathcal{E}_i^r] = \mathbf{Pr}[N_i^r = 0] \approx \frac{\lambda^0 e^{-\lambda}}{0!} = e^{-r/n}. \tag{3.3}$$

The main benefit in using the Poisson approximation is that now we can claim that the events $\mathcal{E}_i^r$, for $1 \leq i \leq n$, are "almost independent," even though it is quite easy to see that there is indeed some dependence between these events. In particular, we make the following informal claim to complete the heuristic calculation.

Claim: For $1 \leq i \leq n$, and for any set of indices $\{j_1, \ldots, j_k\}$ not containing i,

$$\mathbf{Pr}[\mathcal{E}_i^r \mid \cap_{l=1}^k \mathcal{E}_{j_l}^r] \approx \mathbf{Pr}[\mathcal{E}_i^r].$$

PROOF: The proof follows from the following approximate calculations,

$$\begin{aligned}
\mathbf{Pr}[\mathcal{E}_i^r \mid \cap_{l=1}^k \mathcal{E}_{j_l}^r] &= \frac{\mathbf{Pr}[\mathcal{E}_i^r \cap (\cap_{l=1}^k \mathcal{E}_{j_l}^r)]}{\mathbf{Pr}[\cap_{l=1}^k \mathcal{E}_{j_l}^r]} \\
&= \frac{\left(1 - \frac{k+1}{n}\right)^r}{\left(1 - \frac{k}{n}\right)^r} \\
&\approx \frac{e^{-r(k+1)/n}}{e^{-rk/n}} \\
&= e^{-r/n}.
\end{aligned}$$

The first line follows from the definition of conditional expectation (Definition C.4), the second from an elementary probability calculation, and the third from Proposition B.3 (Appendix B). Since the last expression is the approximate value of $\mathbf{Pr}[\mathcal{E}_i^r]$, we obtain the desired result. □

If the approximation in (3.3) were exact, we would obtain that the events $\mathcal{E}_i^r$ are truly independent (Appendix C). In the following computation, we make the heuristic assumption of independence based on the approximation of (3.3). We then obtain that for $1 \leq i \leq n$, the probability that all coupon types are collected in the first m trials is given by:

$$\mathbf{Pr}[\neg(\cup_{i=1}^n \mathcal{E}_i^m)] = \mathbf{Pr}[\cap_{i=1}^n (\neg \mathcal{E}_i^m)] \approx (1 - e^{-m/n})^n \approx e^{-ne^{-m/n}}.$$

Let $m = n(\ln n + c)$ for any constant $c \in \mathbb{R}$. Then, by the preceding argument, we obtain that

$$\begin{aligned}
\mathbf{Pr}[X > m = n(\ln n + c)] &= \mathbf{Pr}[\cup_{i=1}^n \mathcal{E}_i^m] \\
&\approx \mathbf{Pr}[\cap_{i=1}^n (\neg \mathcal{E}_i^m)] \\
&= 1 - e^{-e^{-c}}.
\end{aligned}$$

Observe that this probability $e^{-e^{-c}}$ is close to 1 for large positive c, and is negligibly small for large negative c. Thus, the probability of having collected all n coupon types abruptly changes from nearly zero to almost one in a small interval centered around $n \ln n$. Of course, all this is contingent on our heuristic estimates being close to the true values. The power of this Poisson heuristic is that it gives a quick back-of-the-envelope type estimation of probabilistic quantities, which hopefully provides some insight into the true behavior of those quantities. As we will see in Section 3.6.3, a more rigorous but cumbersome argument can often be used to justify the conclusions obtained from such heuristic arguments.

3.6.3. A Sharp Threshold

We now convert the heuristic argument from the previous section into a rigorous (but significantly more complex) proof using the Boole-Bonferroni Inequalities (Proposition C.2). But first we prove the following technical lemma.

Lemma 3.7: *Let c be a real constant, and $m = n \ln n + cn$ for positive integer n. Then, for any fixed positive integer k,*

$$\lim_{n\to\infty} \binom{n}{k} \left(1 - \frac{k}{n}\right)^m = \frac{e^{-ck}}{k!}.$$

PROOF: Using Proposition B.3.2, we have that

$$e^{\frac{-km}{n}} \left(1 - \frac{k^2}{n}\right)^{\frac{m}{n}} \leq \left(1 - \frac{k}{n}\right)^m \leq e^{\frac{-km}{n}}.$$

Observe that $e^{-km/n} = n^{-k}e^{-ck}$. Further,

$$\lim_{n\to\infty} \left(1 - \frac{k^2}{n}\right)^{\frac{m}{n}} = 1$$

and (by Proposition B.2),

$$\lim_{n\to\infty} \binom{n}{k} \Big/ \frac{n^k}{k!} = 1\,.$$

Putting all this together yields the desired result. □

Theorem 3.8: *Let the random variable X denote the number of trials for collecting each of the n types of coupons. Then, for any constant $c \in \mathbb{R}$, and $m = n \ln n + cn$,*

$$\lim_{n\to\infty} \mathbf{Pr}[X > m] = 1 - e^{-e^{-c}}.$$

PROOF: We have that the event $\{X > m\} = \cup_{i=1}^{n} \mathcal{E}_i^m$. By the Principle of Inclusion-Exclusion,

$$\mathbf{Pr}[\cup_i \mathcal{E}_i^m] = \sum_{k=1}^{n} (-1)^{k+1} P_k^n$$

where

$$P_k^n \triangleq \sum_{1 \leq i_1 < i_2 < \cdots < i_k \leq n} \mathbf{Pr}[\cap_{j=1}^{k} \mathcal{E}_{i_j}^m].$$

Let $S_k^n = P_1^n - P_2^n + P_3^n - \cdots + (-1)^{k+1} P_k^n$ denote the partial sum formed by the first k terms of this series. By the Boole-Bonferroni inequalities (Proposition C.2), we have the bracketing property of the partial sums:

$$S_{2k}^n \leq \mathbf{Pr}[\cup_i \mathcal{E}_i^m] \leq S_{2k+1}^n.$$

By symmetry, all the k-wise intersections of the events $\mathcal{E}_i^m$ are equally likely. This implies that

$$P_k^n = \binom{n}{k} \mathbf{Pr}[\cap_{i=1}^k \mathcal{E}_i^m].$$

Moreover, the probability of the intersection of the k events $\mathcal{E}_1^m, \ldots, \mathcal{E}_k^m$ is the probability of not collecting any of the first k coupons in m trials, namely $(1 - k/n)^m$. Therefore

$$P_k^n = \binom{n}{k} \left(1 - \frac{k}{n}\right)^m.$$

For all positive integers k, define $P_k = e^{-ck}/k!$. By Lemma 3.7 we have that for each k

$$\lim_{n \to \infty} P_k^n = P_k.$$

Define the partial sums of the terms P_k as

$$S_k = \sum_{j=1}^{k} (-1)^{j+1} P_j = \sum_{j=1}^{k} (-1)^{j+1} \frac{e^{-cj}}{j!}.$$

Notice that the right-hand side consists precisely of the first k terms of the power series expansion of $f(c) = 1 - e^{-e^{-c}}$. We conclude that

$$\lim_{k \to \infty} S_k = f(c).$$

That is, for all $\epsilon > 0$, there exists $k^* > 0$ such that for any $k > k^*$,

$$|S_k - f(c)| < \epsilon.$$

Since $\lim_{n \to \infty} P_k^n = P_k$, it follows that $\lim_{n \to \infty} S_k^n = S_k$. Equivalently, for all $\epsilon > 0$ and k, when n is sufficiently large, $|S_k^n - S_k| < \epsilon$. Thus, for all $\epsilon > 0$, any fixed $k > k^*$, and n sufficiently large,

$$|S_k^n - S_k| < \epsilon \text{ and } |S_k - f(c)| < \epsilon,$$

which implies that

$$|S_k^n - f(c)| < 2\epsilon$$

and that

$$|S_{2k}^n - S_{2k+1}^n| < 4\epsilon.$$

Using the bracketing property of partial sums, we obtain that for any $\epsilon > 0$ and n sufficiently large,

$$|\mathbf{Pr}[\cup_i \mathcal{E}_i^m] - f(c)| < 4\epsilon.$$

This implies the desired result that

$$\lim_{n \to \infty} \mathbf{Pr}[\cup_i \mathcal{E}_i^m] = f(c) = 1 - e^{-e^{-c}}.$$

□

By this theorem, for any real constant c, we have

$$\lim_{n\to\infty} \mathbf{Pr}[X \leq n(\ln n - c)] = e^{-e^c}$$

and

$$\lim_{n\to\infty} \mathbf{Pr}[X \geq n(\ln n + c)] = 1 - e^{-e^{-c}}.$$

Thus, we obtain that

$$\lim_{n\to\infty} \mathbf{Pr}[n(\ln n - c) \leq X \leq n(\ln n + c)] = e^{-e^{-c}} - e^{-e^c}.$$

As the value of c is increased, it can be verified that this probability rapidly approaches 1. In other words, with extremely high probability, the number of trials for collecting all n coupon types lies in a small interval centered about its expected value. This result is *almost* like a deterministic result since it so sharply identifies the threshold value for collecting all coupons. We refer to such results as *sharp threshold* results.

Notes

Comprehensive treatises on occupancy problems are the books by Johnson and Kotz [222], and by Kolchin, Chistiakov, and Sevastianov [266]. However, most of the results in these books concern the behavior of the distributions of various random variables in the limit as n becomes large. (See also the various discussions of occupancy problems in the books by Feller [142, 143].) Generally, we will be concerned with statements resembling the ones in Section 3.1, involving asymptotic estimates on random variables and probabilities. We will return to such estimates for occupancy problems in Chapter 4. Recent work by Azar, Broder, Karlin, and Upfal [35] builds on the basic occupancy problem and points out many applications to computer science.

The history of tail inequalities such as the Chebyshev bound dates back to the early days of probability theory. Following Chebyshev's bound [394], Markov [293] observed that the same idea could be used with higher moments. Kolmogorov [267] went further and remarked that $\mathbf{Pr}[X \geq r] \leq \mathbf{E}[f(X)]/s$ for any function $f(X)$, provided that $\mathbf{E}[f(X)]$ exists and $f(x) \geq s > 0$ for all $x \geq r$. The latter idea was exploited by Bernstein and by Chernoff in a manner we will describe in Chapter 4.

Classic sources for deterministic selection algorithms are the papers of Blum, Floyd, Pratt, Rivest, and Tarjan [65], and of Schönhage, Paterson, and Pippenger [364]. The **LazySelect** algorithm presented here is a variant on one reported by Floyd and Rivest [151]. The algorithm described therein is a recursive algorithm, and does not sort after the first level of random sampling as we do. The lower bound of $2n$ for median selection is due to Bent and John [54].

The construction of pairwise independent random variables in Exercise 3.7 is given in Joffe [214]. Its application to the reduction of random bits used by abstract randomized algorithms is due to Chor and Goldreich [97]; Luby [282] presented this idea in the context of a concrete problem we will study in Chapter 12. The two-point sampling technique has been developed into a powerful technique for reducing the use of randomness, especially for the *derandomization* of algorithms (see the Notes section of Chapter 12).

The Proposal Algorithm for stable marriages is due to Gale and Shapley [161]. The book by Gusfield and Irving [188] provides a comprehensive treatment of results related

to stable marriages. Our presentation of the average-case analysis of the Proposal Algorithm is drawn from Knuth's monograph [263]. The power and applicability of the Poisson heuristic is explored in great detail in the monograph by Aldous [12].

Problems

3.1 Consider an occupancy problem in which n balls are independently and uniformly distributed in n bins. Show that, for large n, the expected number of empty bins approaches n/e, where e is the base of the natural logarithm. What is the expected number of empty bins when m balls are thrown into n bins? (See Theorem 4.18.)

3.2 Suppose m balls are thrown into n bins. Give the best bound you can on m to ensure that the probability of there being a bin containing at least two balls is at least 1/2.

3.3 A parallel computer consists of n processors and n memory modules. During a step, each processor sends a memory request to one of the memory modules. A memory module that receives either one or two requests can satisfy its request(s); modules that receive more than two requests will satisfy two requests and discard the rest.

(a) Assuming that each processor chooses a memory module independently and uniformly at random, what is the expected number of processors whose requests are satisfied? Use the approximation $(1 - 1/n)^n \approx 1/e$ if necessary.

(b) Repeat the computation for the case where each memory module can satisfy only one request during a step.

3.4 Consider the following experiment, which proceeds in a sequence of *rounds*. For the first round, we have n balls, which are thrown independently and uniformly at random into n bins. After round i, for $i \geq 1$, we discard every ball that fell into a bin by itself in round i. The remaining balls are retained for round $i + 1$, in which they are thrown independently and uniformly at random into the n bins. Show that there is a constant c such that with probability $1 - o(1)$, the number of rounds is at most $c \log \log n$.

3.5 Let X be a random variable with expectation μ_X and standard deviation σ_X.

(a) Show that for any $t \in \mathbb{R}^+$,

$$\mathbf{Pr}[X - \mu_X \geq t\sigma_X] \leq \frac{1}{1 + t^2}.$$

This version of the Chebyshev inequality is sometimes referred to as the **Chebyshev-Cantelli bound**.

(b) Prove that

$$\mathbf{Pr}[|X - \mu_X| \geq t\sigma_X] \leq \frac{2}{1 + t^2}.$$

Under what circumstances does this give a better bound than the Chebyshev inequality?

3.6 Let Y be a non-negative integer-valued random variable with positive expectation. Prove the following inequalities.

(a)

$$\mathbf{Pr}[Y = 0] \leq \frac{\mathbf{E}[Y^2] - \mathbf{E}[Y]^2}{\mathbf{E}[Y]^2}$$

(b)

$$\frac{\mathbf{E}[Y]^2}{\mathbf{E}[Y^2]} \leq \mathbf{Pr}[Y \neq 0] \leq \mathbf{E}[Y]$$

(c) Explain why the second inequality always gives a stronger bound than the first inequality.

3.7 Let a and b be chosen independently and uniformly at random from $\mathbb{Z}_n = \{0, 1, 2, \ldots, n-1\}$, where n is a prime. Suppose we generate t pseudo-random numbers from $\mathbb{Z}_n$ by choosing $r_i = ai + b \bmod n$, for $1 \leq i \leq t$. For any $\epsilon \in [0, 1]$, show that there is a choice of the witness set $W \subset \mathbb{Z}_n$ such that $|W| \geq \epsilon n$ and the probability that *none* of the r_i's lie in the set W is at least $(1-\epsilon)^2/4t$.

3.8 Suggest a scheme for "four-point" sampling from the range $\mathbb{Z}_n$ where n is a prime. For $t < n$ samples $r_1, \ldots, r_t$ using this scheme, give an upper bound on the probability that all t attempts fail to discover a witness given $x \in L$ and compare this with the bound of $1/16$ that the naive use of four samples would yield. En route, derive an upper bound on the fourth central moment of the sum of four-way independent random variables.

3.9 (Due to D.R. Karger and R. Motwani [233].)
(a) Let S, T be two disjoint subsets of a universe U such that $|S| = |T| = n$. Suppose we select a random set $R \subseteq U$ by independently sampling each element of U with probability p. We say that the random sample R is *good* if the following two conditions hold: $R \cap S = \emptyset$ and $R \cap T \neq \emptyset$. Show that for $p = 1/n$, the probability that R is good is larger than some positive constant.

(b) Suppose now that the random set R is chosen by sampling the elements of U with only *pairwise* independence. Show that for a suitable choice of the value of p, the probability that R is good is larger than some positive constant.

3.10 The sharp threshold result in the coupon collector's problem does not imply that the probability of needing more than $cn \log n$ trials goes to zero at a doubly exponential rate if c were not a constant, but were allowed to grow with n. Let the probability of requiring more than $cn \log n$ trials be $p(c)$. For constant c, show that $1/p(c)$ can be bounded from above and below by polynomials in n.

3.11 Consider the extension of the coupon collector's problem to that of collecting at least k copies of each coupon type. Show that the sharp threshold for the number of selections required (denoted $X^{(k)}$) is centered at $n(\ln n + (k-1)\ln\ln n)$. In other words, for any positive integer k and constant $c \in \mathbb{R}$, prove that

$$\lim_{n \to \infty} \mathbf{Pr}[X^{(k)} > n(\ln n + (k-1)\ln\ln n + c)] = e^{-e^{-c}}.$$

3.12 Consider the following process related to the coupon collector problem. There are n bins and n players, and each player has an infinite supply of balls. The bins are all initially empty. We have a sequence of rounds: in each round, each player throws a ball into an empty bin chosen independently at random from all currently empty bins. Let the random variable Z be the number of rounds before every bin is non-empty. Determine the expected value of Z. What can you say about the tail of Z's distribution?

3.13 Let B be a random bipartite graph on two independent sets of vertices U and V, each with n vertices. For each pair of vertices $u \in U$ and $v \in V$, the probability that the edge between them is present is $p(n)$, and the presence of any edge is independent of all other edges. Let $p(n) = (\ln n + c)/n$ for some $c \in \mathbb{R}$.

(a) Show that the probability that B contains an isolated vertex is asymptotically equal to $e^{-2e^{-c}}$.

(b) Suggest and prove a generalization of this to random non-bipartite graphs.

3.14 (Due to R.M. Karp.) Consider a bin containing d balls chosen at random (without replacement) from a collection of n distinct balls. Without being able to see or count the balls in the bin, we would like to simulate random sampling *with replacement* from the original set of n balls. Our only access to the balls is that we can sample *without replacement* from the bin.

Consider the following strategy. Suppose that $k < d$ balls have been drawn from the bin so far. Flip a coin with the probability of HEADS being k/n. If HEADS appears, then pick one of the k previously drawn balls uniformly at random; otherwise, draw a random ball from the bin. Show that each choice is independently and uniformly distributed over the space of the n original balls. How many times can we repeat the sampling?

3.15 (Due to D. Angluin and L.G. Valiant [28].) Let B denote a random bipartite graph with n vertices in each of the vertex sets U and V. Each possible edge, independently, is present with probability $p(n)$. Consider the following algorithm for constructing a perfect matching (see Section 7.3) in such a random graph. Modify the Proposal Algorithm of Section 3.5 as follows. Each $u \in U$ can propose only to adjacent $v \in V$. A vertex $v \in V$ always accepts a proposal, and if a proposal causes a "divorce," then the newly divorced $u \in U$ is the next to propose. The sampling procedure outlined in Problem 3.14 helps implement the Principle of Deferred Decisions. How small can you make the value of $p(n)$ and still have the algorithm succeed with high probability? The following fact concerning the degree $d(v)$ of a vertex v in B proves useful:

$$\mathbf{Pr}[d(v) \leq (1-\beta)np] = O\left(e^{-\beta^2 np/2}\right).$$