Lecture 9

Randomized Algorithms II

Supplemental reading in CLRS: Appendix C; Section 7.3

After Lecture 8, several students asked whether it was fair to compare randomized algorithms to deterministic algorithms.

Determistic algorithm:
- always outputs the right answer
- always runs efficiently

Randomized algorithm:
- may sometimes not output the right answer
- may sometimes not run efficiently

Is this "fair"? Of course not. We demand less from randomized algorithms. But in exchange, we expect randomized algorithms to do more—to be more efficient, or to be simple and elegant.

Separately, we might ask whether it is *useful* to employ randomized algorithms. The answer here is that it depends:

- on the situation
- on the available alternatives
- on the probability of error or inefficiency.

Lots of real-life situations call for randomized algorithms. For example, Google Search and IBM's *Jeopardy!*-playing computer Watson both employ randomized algorithms.

9.1 The Central Limit Theorem and the Chernoff Bound

Ideally, a randomized algorithm will have the property that

$$\Pr\left[\text{bad things happening}\right] \xrightarrow{n\to\infty} 0.$$

Before further analyzing randomized algorithms, it will be useful to establish two basic facts about probability theory: the central limit theorem and the Chernoff bound. The central limit theorem states that the mean of a large number of independent copies of a random variable is approximately normal, as long as the variable has finite variance. The **normal distribution** (or **Gaussian distribution**) of mean 0 and variance 1, denoted $N(0,1)$, is defined by the probability density function

$$f_{N(0,1)}(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-x^2/2\right).$$

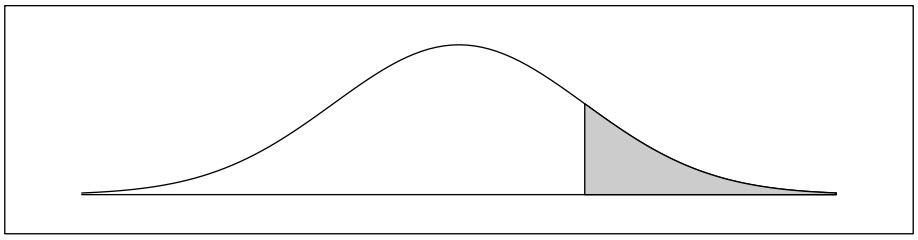

Figure 9.1. A tail of the normal distribution.

Taking an affine transform of this standard Gaussian, we obtain the normal distribution with arbitrary mean μ and arbitrary variance σ^2, denoted $N(\mu,\sigma^2)$. Its probability density function is

$$f_{N(\mu,\sigma^2)}(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\ \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

Due in part to the following theorem, the Gaussian distribution is extremely important in all of probability theory.

Theorem 9.1 (Central Limit Theorem)**.** *Suppose $X_1, X_2, \ldots$ are i.i.d.*[1] *random variables, each with finite mean μ and finite variance σ^2. Let $S_n = \frac{X_1+\ldots+X_n}{n}$, and let $Y_n = \sqrt{n}(S_n - \mu)$. Then the variables $Y_1, Y_2, \ldots$ converge to a normal distribution:*

$$Y_n \xrightarrow{d} N(0,\sigma^2).$$

The precise meaning of this theorem is somewhat technical. Again, in effect it means that the average of a large number of independent copies of a random variable X is approximately normally distributed. Thus, the probability that this average exceeds its expected value by r is approximately the probability that a normal random variable with variance σ^2 exceeds its expected value by r. While this heuristic reasoning does not rigorously prove any specific bound, there are a number of bounds that have been worked out for certain common distributions of X. One such bound is the Chernoff bound, one version of which is as follows:

Theorem 9.2 (Chernoff bound)**.** *Let $Y \sim B(n,p)$ be a random variable representing the total number of heads in a series of n independent coin flips, where each flip has probability p of coming up heads. Then, for all $r > 0$, we have*

$$\Pr\left[Y \geq \mathbb{E}[Y] + r\right] \leq \exp\left(-2r^2/n\right).$$

The distribution in Theorem 9.2 is called the **binomial distribution** with parameters (n,p). The probability of m heads is $\binom{n}{m}p^m(1-p)^{n-m}$.

There are several different versions of the Chernoff bound—some with better bounds for specific hypotheses, some more general. Section C.5 of CLRS gives a classic proof of the Chernoff bound, applying the Markov inequality to the random variable $\exp\left(\alpha(Y - \mathbb{E}[Y])\right)$ for a suitable constant α. We give a different proof here[2], due to Impagliazzo and Kabanets, 2010.

[1] independent and identically distributed

[2] Of a different theorem, actually. The version we prove here is (stronger than) Exercise C.5-6 of CLRS.

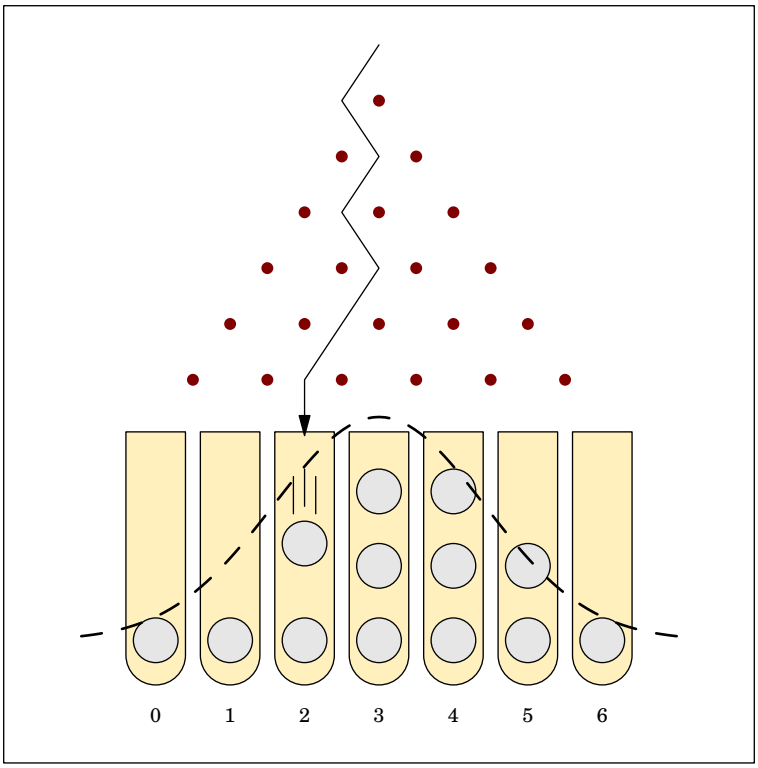

Figure 9.2. *Galton's board* is a toy in which, at each peg, the ball has probability $\frac{1}{2}$ of moving left and probability $\frac{1}{2}$ of moving right. The number of the bin into which it falls is a binomial random variable with parameters $(n, \frac{1}{2})$. If n is large and you drop a large number of balls into the board, the collection of balls in the bins underneath will start to look like a Gaussian probability density curve.

Proof of Theorem 9.2. Let

$$Y = X_1 + \cdots + X_n,$$

where $X_1, \ldots, X_n$ are independent $\{0,1\}$-valued random variables, each having probability p of being 1. Thus Y is a binomial random variable with parameters (n,p). Let S be a random subset of $\{1,\ldots,n\}$ such that each element of $\{1,\ldots,n\}$ independently has probability q of being included in S, where q is a parameter that will be determined later. The probability that $X_i = 1$ for all $i \in S$ is

$$\begin{aligned}\Pr\left[\bigwedge_{i\in S} X_i = 1\right] &= \Pr\left[\bigwedge_{i\in S} X_i = 1 \;\middle|\; Y \geq \mathbb{E}[Y] + r\right] \Pr\Big[Y \geq \mathbb{E}[Y] + r\Big] \\ &\quad + \Pr\left[\bigwedge_{i\in S} \;\middle|\; Y < \mathbb{E}[Y] + r\right] \Pr\Big[Y < \mathbb{E}[Y] + r\Big] \\ &\geq \Pr\left[\bigwedge_{i\in S} X_i = 1 \;\middle|\; Y \geq \mathbb{E}[Y] + r\right] \Pr\Big[Y \geq \mathbb{E}[Y] + r\Big]. \qquad (9.1)\end{aligned}$$

Meanwhile,

$$\Pr\left[\bigwedge_{i\in S} X_i = 1 \;\middle|\; Y \geq \mathbb{E}[Y] + r\right] \geq (1-q)^{n-(\mathbb{E}[Y]+r)}, \qquad (9.2)$$

since the condition $\bigwedge_{i\in S} X_i = 1$ is equivalent to the condition that S doesn't include the indices of any zeros, and we are conditioning on the hypothesis that there are at most $n - (\mathbb{E}[Y] + r)$ zeros. (This is the "key idea" of the proof. Notice that we have switched from conditioning on S followed by the X_i's to conditioning on the X_i's followed by S.) Combining (9.1) and (9.2), we obtain

$$\Pr\left[\bigwedge_{i\in S} X_i = 1\right] \geq (1-q)^{n-(\mathbb{E}[Y]+r)} \Pr\Big[Y \geq \mathbb{E}[Y] + r\Big]. \qquad (9.3)$$

We could also compute $\Pr\left[\bigwedge_{i\in S} X_i = 1\right]$ by conditioning on S in a different way:

$$\Pr\left[\bigwedge_{i\in S} X_i = 1\right] = \sum_{k=0}^{n} \underbrace{\left(\tbinom{n}{k} q^k (1-q)^{n-k}\right)}_{\Pr\left[|S|=k\right]} \cdot \underbrace{\left(\qquad p^k \qquad\right)}_{\Pr\left[\bigwedge_{i\in S} X_i = 1 \;\middle|\; |S|=k\right]}.$$

Recall the binomial formula, which states

$$(a+b)^n = \sum_{k=0}^{n} \tbinom{n}{k} a^k b^{n-k}.$$

In the present case, we have

$$\Pr\left[\bigwedge_{i\in S} X_i = 1\right] = \big(qp + 1 - q\big)^n. \qquad (9.4)$$

Combining (9.3) and (9.4), we obtain

$$(1-q)^{n-(\mathbb{E}[Y]+r)} \Pr\Big[Y \geq \mathbb{E}[Y] + r\Big] \leq \big(qp + 1 - q\big)^n.$$

Rearranging and using the fact that $\mathbb{E}[Y] = np$, we have equivalently

$$\Pr\Big[Y \geq \mathbb{E}[Y] + r\Big] \leq \frac{\big(qp + 1 - q\big)^n}{(1-q)^{n-(np+r)}}. \qquad (9.5)$$

Equation (9.5) holds for arbitrary q, so we might as well assign a value to q for which the right side is minimal. Let

$$q = \frac{r/n}{(p+\frac{r}{n})(1-p)};$$

after some expansion, the right side of (9.5) becomes

$$\left[\left(\frac{p}{p+\frac{r}{n}}\right)^{p+\frac{r}{n}}\left(\frac{1-p}{1-(p+\frac{r}{n})}\right)^{1-(p+\frac{r}{n})}\right]^n = \exp\left(-nf\left(\tfrac{r}{n}\right)\right),$$

where[3]

$$f(x) = (p+x)\ln\left(1+\tfrac{x}{p}\right) + (1-(p+x))\ln\left(1-\tfrac{x}{1-p}\right).$$

Below we will show that

$$f(x) \geq 2x^2 \quad \text{for all } x \in \mathbb{R} \text{ such that } f(x) \in \mathbb{R},$$

so that (9.5) becomes

$$\begin{aligned}\Pr\left[Y \geq \mathbb{E}[Y] + r\right] &\leq \exp\left(-nf\left(\tfrac{r}{n}\right)\right)\\ &\leq \exp\left(-2n\left(\tfrac{r}{n}\right)^2\right)\\ &= \exp\left(-\tfrac{2r^2}{n}\right).\end{aligned}$$

This will complete the proof.

All that remains is to show that $f(x) \geq 2x^2$ for all x such that $f(x) \in \mathbb{R}$—namely, for $-p < x < 1-p$, though we will only need the range $0 \leq x < 1-p$.[4] This can be shown by calculus: the first and second derivatives of f are

$$\begin{aligned}f'(x) &= \ln\left(1+\tfrac{x}{p}\right) - \ln\left(1-\tfrac{x}{1-p}\right),\\ f''(x) &= \frac{1}{(1-(p+x))(p+x)}.\end{aligned}$$

In particular, $f''(x)$ is minimized when $p+x = \frac{1}{2}$, at which point $f''(x) = 4$. Thus, the fundamental theorem of calculus gives

$$f'(x) = \underbrace{f'(0)}_{0} + \int_{t=0}^{x} f''(t)\,dt \geq \int_{t=0}^{x} 4\,dt = 4x.$$

Applying the fundamental theorem of calculus again, we have

$$f(x) = \underbrace{f(0)}_{0} + \int_{t=0}^{x} f'(t)\,dt \geq \int_{t=0}^{x} 4t\,dt = 2x^2. \qquad \square$$

[3] The definition of f may seem unmotivated here, but in fact $f(x)$ is properly interpreted as the *relative entropy* between two Bernoulli random variables with parameters p and $p+x$, respectively. Relative entropy is a measure of the "distance" between two probability distributions.

[4] Note that $x = \frac{r}{n}$ will always fall within this range, as we must have $\mathbb{E}[Y] + r \leq n$. (Or at least, the theorem becomes trivial when $\frac{r}{n} > 1-p$.)

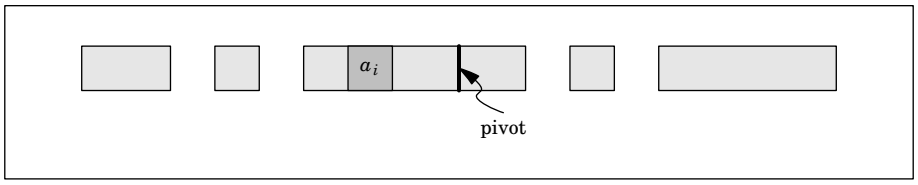

Figure 9.3. In a given iteration of QUICKSORT, there will be several working subarrays, each in the process of being sorted by its own recursive call to QUICKSORT.

9.2 Analysis of QUICKSORT

Recall the procedure QUICKSORT, which sorts an array $A = \langle a_1, \ldots, a_n \rangle$ in place (see Figure 9.3):

Algorithm: QUICKSORT(A)

1. Choose an element $x \in A$ uniformly at random. We will call x the "pivot."
2. Partition A around x.
3. Sort the elements less than x recursively.
4. Sort the elements greater than x recursively.

We saw in recitation that the expected running time of QUICKSORT is $\Theta(n \lg n)$. In this lecture we'll use the Chernoff bound to show that, for some constant $c \geq 1$, the probability that QUICKSORT takes more than $cn \lg n$ time on any given input is at most $\frac{1}{n}$. In fact, what we'll show is that after time $cn \lg n$, the probability that the working subarray containing any given element $a \in A$ consists of more than one element is at most $\frac{1}{n^2}$. Once this is established, the **union bound** shows that the probability that there exists *any* working subarray with more than one element is at most $\frac{1}{n}$. That is to say, let $A = \langle a_1, \ldots, a_n \rangle$ and let E_i be the event that after time $cn \lg n$, the working subarray containing a_i has more than one element. Then the union bound states that

$$\Pr\left[\begin{array}{c}\exists \text{ working subarray with} \\ \text{more than one element}\end{array}\right] = \Pr\left[\bigcup_{i=1}^{n} E_i\right] \leq \sum_{i=1}^{n} \Pr[E_i] \leq n\left(\frac{1}{n^2}\right) = \frac{1}{n}.$$

(In general, the union bound says that the probability of a finite or countably infinite union of events E_i is at most the sum of their individual probabilities. The intuition is that equality is achieved when the E_i's are pairwise disjoint, and any overlap only diminishes the size of the union.)

All that remains is to find c such that, for any array A,

$$\Pr[E_i] \leq \frac{1}{n^2} \quad \text{for all } i = 1, \ldots, n.$$

As in Lecture 8, we define a "good" pivot x in an array A to be a pivot such that

$$\tfrac{1}{10}|A| \leq \text{rank}(x) \leq \tfrac{9}{10}|A|.$$

Thus, a good pivot splits the current array into two subarrays each at most $\frac{9}{10}$ as big. Let $X_{i,k}$ be the indicator random variable

$$X_{i,k} = \begin{cases} 0 & \text{if, in the } k\text{th iteration, a good pivot was chosen for } a_i\text{'s subarray} \\ 1 & \text{otherwise.} \end{cases}$$

(By "the kth iteration," we mean all calls to QUICKSORT at recursive depth k. Thus, the kth iteration will consist of 2^{k-1} recursive calls to QUICKSORT, assuming none of the subarrays have shrunk to size 1 yet.) Note two things:

- $\mathbb{E}\left[X_{i,k}\right] = 0.2$ for all i and all k. (To reduce the amount of bookkeeping we have to do, we will assume that pivots continue to be chosen even after an element's subarray is reduced to size 1, and that the probability of choosing a good pivot continues to be 0.8.)
- For a fixed k, the variables $X_{1,k}, X_{2,k}, \ldots, X_{n,k}$ are dependent on each other. However, for a fixed i, the variables $X_{i,1}, X_{i,2}, \ldots$ are independent of each other.

For a fixed i and a fixed K, we have

$$\mathbb{E}\left[\sum_{k=1}^{K} X_{i,k}\right] = 0.2K.$$

Thus, by the Chernoff bound (with $r = 0.1K$), we have

$$\Pr\left[\sum_{k=1}^{K} X_{i,k} > 0.2K + 0.1K\right] \leq \exp\left(-\frac{2(0.1K)^2}{K}\right) = \exp(-0.02K).$$

Now let $K = 100\ln n$. We obtain

$$\Pr\left[\sum_{k=1}^{K} X_{i,k} > 0.3K\right] \leq \frac{1}{n^2}.$$

On the other hand, if $\sum_{k=1}^{K} X_{i,k} \leq 0.3K$, then at least $0.7K$ of the first K pivots were good, and the size of the working subarray containing a_i after K steps is at most

$$n \cdot \left(\frac{9}{10}\right)^{0.7K} = n \cdot \left(\frac{9}{10}\right)^{70\ln n} = n^{70\ln(9/10)+1} \approx n^{-6.3} < 1$$

(i.e., the size is at most 1; the reason for the fractional number is that we are ignoring issues of rounding). Thus, after $K = 100\ln n$ iterations, the probability that the working subarray containing a_i has more than one element is at most $\frac{1}{n^2}$. So by the union bound, the probability that QUICKSORT requires more than K iterations is at most $\frac{1}{n}$.

While the sizes of the inputs to each iteration and the total number of iterations required are random, the running time of the non-recursive part of QUICKSORT is deterministic: It takes $\Theta(A)$ time to pick a random element and partition around that element. Let's say it takes at most τA time to do this, where τ is some appropriately chosen constant. Now, if the sizes of the inputs to the kth iteration are $m_1, \ldots, m_\ell$, then the quantities $m_1, \ldots, m_\ell$ are random, but we know $\sum_{\lambda=1}^{\ell} m_\lambda = n$, so the total amount of time required by the kth iteration is at most $\sum_{\lambda=1}^{\ell} \tau m_\lambda = \tau n$. Thus, the probability that QUICKSORT takes more than $K\tau n = (100\tau) n \ln n$ time is at most $1/n$. This is what we set out to show in the beginning of this section, with $c = 100\tau$.

To recap:

> How can we argue about randomized algorithms?
>
> - Identify the random choices in the algorithm.
> - After fixing the random choices, we have a deterministic algorithm.

In this example, the random choices were the pivots. Fixing the sequence of choices of pivots, we were able to analyze $|A_{i,k}|$, the size of the subarray containing a_i after k iterations, for each i and k (see Figure 9.4).

Random choices at depth 1	$\lvert A_{1,1}\rvert = n$	$\lvert A_{2,1}\rvert = n$	$\cdots$
Random choices at depth 2	$\lvert A_{1,2}\rvert = 0.3n$	$\lvert A_{2,2}\rvert = 0.8n$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$

Figure 9.4. The values $0.3n$ and $0.8n$ are just examples, and of course depend on the sequence of random choices. Once the sequence of random choices is fixed, we would like to know how many rows down we must go before all entries are 1.

9.3 Monte Carlo Sampling

Suppose that out of a large population U, there occurs a certain phenomenon in a subset $S \subseteq U$. Given an element x, we are able to test whether $x \in S$. How can we efficiently estimate $\frac{|S|}{|U|}$? For example, how can we efficiently estimate what fraction of the world's population is left-handed, or what percentage of voters vote Republican?

There is an elegant simple solution to this problem: Choose a k-element sample $A = \{x_1, \ldots, x_k\} \subseteq U$ uniformly at random. We then estimate $\frac{|S|}{|U|}$ by the quantity $\widehat{S} = \frac{|A \cap S|}{|A|} = \frac{1}{k}\left|\{i : x_i \in S\}\right|$, figuring that the chance that an element $a \in A$ belongs to S ought to be the same as the chance that a general element $x \in U$ belongs to S. Thus, we are relying heavily on our ability to pick a truly uniform sample.[5] Assuming this reliance is safe, $k\widehat{S} = |A \cap S|$ is a binomial random variable with parameters $\left(k, \frac{|S|}{|U|}\right)$. In particular, the expected value of $\widehat{S}$ is the exact correct answer $\frac{|S|}{|U|}$. Thus, the Chernoff bound states that for any $r > 0$,

$$\Pr\left[\widehat{S} \geq \frac{|S|}{|U|} + r\right] = \Pr\left[k\widehat{S} \geq \mathbb{E}\left[k\widehat{S}\right] + kr\right] \leq \exp\left(-2kr^2\right).$$

Exercise 9.1. *How can we use the Chernoff bound to give an upper bound on* $\Pr\left[\widehat{S} \leq \frac{|S|}{|U|} - r\right]$*? (Hint: replace S by its complement $U \setminus S$.)*

9.4 Amplification

Suppose we are given a Monte Carlo algorithm which always runs in time T and returns the correct answer with probability 2/3. (Assume there is a unique correct answer.) Then, for any $\epsilon > 0$, we can create a new randomized algorithm which always runs in time $O\left(T \lg \frac{1}{\epsilon}\right)$ and outputs an incorrect answer with probability at most ϵ. How?

The plan is to run the original algorithm k times, where $k = O\left(\lg \frac{1}{\epsilon}\right)$ is a quantity which we will determine later. The algorithm should then output whichever answer occurred most frequently out of the k computed answers. To bound the probability of error, let I_j be the indicator random variable which equals 1 if the jth run returns an incorrect answer, and let $I = \sum_{j=1}^{k} I_j$. Then I is a binomial random variable with parameters $\left(k, \frac{1}{3}\right)$. Thus, by the Chernoff bound,

$$\Pr\left[\begin{matrix}\text{we ultimately return}\\ \text{an incorrect answer}\end{matrix}\right] \leq \Pr\left[I \geq \tfrac{1}{2}k\right] = \Pr\left[I \geq \mathbb{E}[I] + \tfrac{1}{6}k\right] \leq \exp\left(-\tfrac{1}{18}k\right).$$

[5] In practice, it could be quite hard to pick a uniform sample. For example, what if you wanted figure out what proportion of New Yorkers speak Chinese? How much of your random sampling should be done in Chinatown? Without the help of extensive census data, it can be hard to make unbiased choices.

Thus, the probability of error will be at most ϵ if we let $k = 18 \ln \frac{1}{\epsilon}$.

Exercise 9.2. *Suppose that instead of returning the correct answer with probability 2/3, our Monte Carlo algorithm returned the correct answer with probability p. What conditions on p allow the above strategy to work? In terms of p and ϵ, how many times must we run the Monte Carlo algorithm?*

6.046J / 18.410J Design and Analysis of Algorithms
Spring 2012