Randomised Algorithms: Expectation and Concentration Bounds

May 29, 2024

1 Basic techniques

Exercise 1 [In expectation vs whp]

(a) Why would one want to use randomness in an algorithm?

(b) Explain the difference between guarantees that come "in expectation" and guarantees that come "with high probability".

(c) Compare quantitatively the outputs of the following python programs:

```
import numpy as np

ans = []
for _ in range(20):
    m = 1
    while np.random.random() < 0.8:
        m += 1
    ans.append(m)
print(ans)
```

and

```
import numpy as np

ans = []
for _ in range(20):
    n = 10000
    m = n
    loads = np.zeros(n)
    for _ in range(m):
        i1 = np.random.randint(0, n - 1)
        i2 = np.random.randint(0, n - 1)
        i_min = i1 if loads[i1] < loads[i2] else i2
        loads[i_min] += 1

    ans.append(np.max(loads) - m / n)
print(ans)
```

Exercise 2 [For sufficiently large n] In several settings, we only care about what happens for sufficiently large n. Convince yourself that the following inequalities hold for sufficiently large n:

- $6n + n^2 \leq 2n^2$,
- $3n^2 - 6n \geq 2n^2$,
- $5n \log^2 n + 3n - 2n \log n \geq 4n \log^2 n$.

Exercise 3 [High probability events combine well] In this exercise, you will see that high probability events combine well. (This may make more sense after you have seen a few examples where we combine high probability events)

(a) Let A and B be two events with $\mathbf{Pr}[A] \geq 1 - n^{-c}$ and $\mathbf{Pr}[B] \geq 1 - n^{-c}$ for some constant $c > 0$. Then $\mathbf{Pr}[A \cap B] \geq 1 - 2n^{-c}$.

(b) How would you interpret in words the result in (a)?

(c) More generally for events $E_1, \ldots, E_m$ with $\mathbf{Pr}[E_i] \geq 1 - n^{-c}$, we have that

$$\mathbf{Pr}\left[\bigcap_{i=1}^{m} E_i\right] \geq 1 - n^c \cdot m.$$

(d) Let A and B be two events such that $\mathbf{Pr}[A \mid B] \geq 1 - n^{-c}$ and $\mathbf{Pr}[B] \geq 1 - n^{-c}$ for some constant $c > 0$. Then, $\mathbf{Pr}[A] \geq 1 - 2n^{-c}$.

(e) How would you interpret in words the result in (d)?

(f) Generalise the result in (d).

Extended Note 1 [Indexed sets] Sometimes you will see $(E_i)_{i=1}^{n}$ instead of $E_1, \ldots, E_n$ or $(E_i)_{i=0}^{\infty}$ instead of $E_1, E_2, \ldots$. This is just for notational convenience.

Exercise 4 [Probability amplification] Consider an algorithm A for a minimisation problem. Further, assume that A succeeds in finding the minimum solution with probability at least $p > 0$, otherwise produces a solution with larger value.
We define the algorithm A' which repeats algorithm A, $t := \lceil \frac{c}{p} \cdot \log n \rceil$ times for some constant c (and n being a parameter of the given problem) and returns the minimum of these values. Then, A' has a success probability of at least $1 - n^{-c}$.

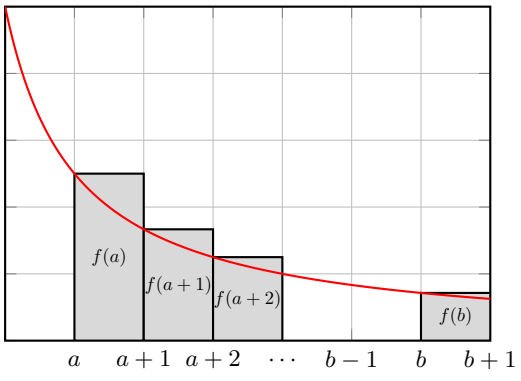
1.1 Common asymptotics

The following exercise demonstrates the general approach for bounding (discrete) sums using integrals. You may find it simpler to first attempt the bound on the harmonic numbers and then come back to this.

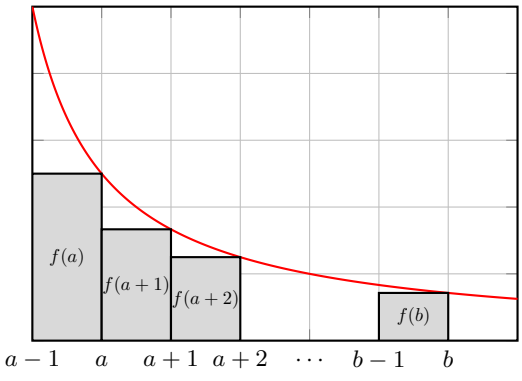
Exercise 5 [General approach] Let $f : [a-1, b+1] \to \mathbb{R}$ be a monotonically decreasing and integrable function for $a, b \in \mathbb{Z}$. Using the two figures below argue that

$$\int_a^{b+1} f(x)\,dx \leq \sum_{i=a}^{b} f(i) \leq \int_{a-1}^{b} f(x)\,dx.$$

Lower bound



Upper bound



Exercise 6 [Harmonic numbers] The n-th *harmonic number* H_n is defined as $H_n = \sum_{i=1}^{n} \frac{1}{i}$. Show that for any $n \geq 1$,

$$\ln n \leq H_n \leq \ln n + 1.$$

Further Reading 1 [Euler–Mascheroni constant] For most applications in algorithms a $\Theta(\log n)$ bound is sufficient for H_n (let alone a $\log n + \Theta(1)$ bound). In case you want to look it up,

$$\lim_{n\to\infty} H_n = \log n + \gamma = \log n + 0.577..,$$

where γ is known as the Euler–Mascheroni constant.

Exercise 7 [Factorial inequalities]

(a) Using Exercise 5, show that

$$n \log n - n + 1 \leq \ln(n!) \leq n \log n - n + 1 + \log n,$$

and deduce that

$$e \cdot \left(\frac{n}{e}\right)^n \leq n! \leq e \cdot n \cdot \left(\frac{n}{e}\right)^n.$$

(b) (optional) By drawing a figure and using the concavity of the $\log(\cdot)$ function, argue that

$$\int_{i-1}^{i} \log x \, dx \geq \frac{\log(i-1) + \log i}{2}.$$

By aggregating over all $i = 1, \ldots, n$ show that

$$n! \leq e\sqrt{n} \cdot \left(\frac{n}{e}\right)^n.$$

Further Reading 2 [Stirling's approximation formula] Stirling's approximation formula says that

$$n! \sim n^n e^{-n} \sqrt{2\pi n},$$

and the following bounds hold for all n,

$$e^{1/(12n+1)} \leq \frac{n!}{n^n e^{-n} \sqrt{2\pi}} \leq e^{1/(12n)}.$$

1.2 Inequalities and Taylor estimates

In the course, you have often used the inequality $1 + x \leq e^x$. In this section you wil prove that this inequality holds using a general (and simple) methodology.

Extended Note 2 Assuming that we want to prove the inequality

$$g(x) \leq f(x),$$

for all values x in some range $[a, b]$.

- **Step 1:** Define the function
$$h(x) := f(x) - g(x).$$
- **Step 2:** Determine the minimum value of this function $h(x^*)$.
- **Step 3:** Deduce that
$$h(x) \geq h(x^*).$$
If $h(x^*) \geq 0$, then the inequality holds. Otherwise, for x^* the inequality fails.

Exercise 8 Prove the inequality $1 + x \leq e^x$ for any $x \in \mathbb{R}$.

Extended Note 3 [Inequalities via Taylor estimates] Another way to derive inequalities for sufficiently small x is via Taylor estimates. For instance,

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots.$$

Then by truncating the series at a certain point, e.g.,

$$e^x \approx 1 + x + \frac{x^2}{2!},$$

we can obtain upper and lower bounds for sufficiently small x by changing the constant in the last term. For instance, we can obtain the inequalities,

$$e^x \leq 1 + x + x^2,$$

and

$$e^x \geq 1 + x + \frac{1}{4}x^2.$$

for any $|x| \leq 1/2$.

Exercise 9 [Birthday paradox] In this exercise, you will prove that with (positive) constant probability you need to take $\Theta(\sqrt{n})$ samples until a collision occurs. Let $\mathcal{E}_i$ be the event that the first collision occurs in the first i samples.

(a) Show that the probability that the first i samples are distinct is

$$\mathbf{Pr}\left[\,\mathcal{E}_i\,\right] = \left(1 - \frac{1}{n}\right) \cdot \left(1 - \frac{2}{n}\right) \cdot \dots \cdot \left(1 - \frac{i-1}{n}\right).$$

(b) Using the inequality $1 + x \leq e^x$ and choosing a constant $C > 0$ appropriately show that

$$\mathbf{Pr}\left[\,\mathcal{E}_{C\sqrt{n}}\,\right] \leq 1/2.$$

(c) Using the inequality $e^x \leq 1 + x + x^2$ (for $x \leq 1$) show that $e^{-u-u^2} \leq 1 - u$ for $u \leq 1/2$.

(d) By choosing a constant $c > 0$ appropriately show that

$$\mathbf{Pr}\left[\,\mathcal{E}_{c\sqrt{n}}\,\right] \geq 1/e.$$

(e) State and deduce the birthday paradox. Why might it not (always) apply in the real-world?

(f) (Maybe) Looking at the crsids of the students that signed up for "Randomised algorithms", what can the birthday paradox say?

Exercise 10 [Lower bounding ratios] In this exercise, we will prove a simple inequality that is useful for bounding ratios, for instance in approximation algorithms.
For any $\epsilon \in (-1, 1)$, it holds that

$$\frac{1}{1+\epsilon} \geq 1 - \epsilon.$$

(a) Prove the inequality.

(b) Consider a maximisation problem P. Assume you have proven that the optimal solution is at most $n^2 + 10n$ and your algorithm returns a solution at least $n^2 - 3n \log n$. Then prove that your algorithm has an approximation ratio of at most $1 + o(1)$.

(c) Repeat the previous exercise when the lower bound is $\frac{1}{2}n^2 - 3n \log n$,

Exercise 11 [Bernoulli's inequality] Prove that for any $n \in \mathbb{N}$ and for any $x \in [-1, +\infty)$, we have that

$$(1+x)^n \geq 1 + nx.$$

1.3 Indicator random variables

Exercise 12 [Basic properties] Let A, B be two events in a sample space Ω. Then, prove the following properties for the indicators of these events:

(a) $\mathbf{E}[\mathbf{1}_A] = \mathbf{Pr}[A]$,

(b) $\mathbf{1}_{A\cap B} = \mathbf{1}_A \cdot \mathbf{1}_B$,

(c) $\mathbf{1}_{\neg A} = 1 - \mathbf{1}_A$,

(d) $\mathbf{1}_{A\cup B} = \mathbf{1}_A + \mathbf{1}_B - \mathbf{1}_{A\cap B}$,

(e) $\mathbf{Var}[\mathbf{1}_A] = \mathbf{Pr}[A] \cdot (1 - \mathbf{Pr}[A])$,

(f) $\mathbf{E}\left[(\mathbf{1}_A)^k\right] = \mathbf{Pr}[A]$ for any $k > 0$.

Exercise 13 [Inequalities]

(a) Argue that for random variables X and Y such that $X \leq Y$ (and have finite expectations), we also have that $\mathbf{E}[X] \leq \mathbf{E}[Y]$.

(b) (**Markov's inequality**) Let X be a non-negative random variable, then show that for any $a > 0$,

$$\mathbf{Pr}[X \geq a] \leq \frac{\mathbf{E}[X]}{a}.$$

Hint: Consider the indicator r.v. of the event $\{X \geq a\}$.

(c) (**Union bound**) Consider n events $\mathcal{E}_1, \ldots, \mathcal{E}_n$ in a sample space Ω. Then,

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

Hint: See **Lecture 1 slide 12**.

Exercise 14 [Fixed points of random permutation] The n passengers of an airplane take a seat uniformly at random.

(a) What is the number of passengers X that sit in their assigned seat in expectation?

(b) What is the variance of X?

(c) (optional +) Assume that n is divisible by 6, the passengers have arrived in $n/2$ pairs and there are $n/6$ rows of seats each with 6 seats. What is the expected number of pairs that will seat next to each other?

Exercise 15 [Inversions in a random permutation] Consider a permutation π of the set $[n] := \{1, \ldots, n\}$. An *inversion* is a pair (i, j) such that $i < j$ and $\pi_i > \pi_j$. Intuitively, it captures the pairs of indices that are out of order.
Show that if π is chosen uniformly at random, then the expected number of inversions N, satisfies

$$\mathbf{E}[N] = \frac{n \cdot (n-1)}{4}.$$

How does this quantity relate to *insertion sort*?

Exercise 16 [Records of a random permutation] Given a random permutation π of the set $[n] := \{1, \ldots, n\}$, how many times does line 5 in the following code execute?

```
1: function FindMin(x_1, ..., x_n)
2:     m ← ∞
3:     for i = 1 to n do
4:         if x_i < m then
5:             m ← x_i
6:         end if
7:     end for
8:     return m
9: end function
```

Exercise 17 [Local max in random permutation] In a sequence $x_1, \ldots, x_n$ a local maximum is an index i such that x_i is at least as large as its (at most two) neighbours x_{i-1} and x_{i+1}.
Compute the expected number of local maxima in expectation for a random permutation of the elements $[n] := \{1, \ldots, n\}$.

Exercise 18 [Number of cycles in random permutation] In this exercise, you will bound the number of cycles in a random permutation of the elements of the set $[n] := \{1, \ldots, n\}$.

(a) Consider a set of distinct elements $\{i_1, \ldots, i_k\}$ of $[n]$. Prove that the probability they form a cycle is
$$\frac{1}{n} \cdot \frac{1}{n-1} \cdot \ldots \cdot \frac{1}{n-k+1}.$$

(b) Prove that the expected number of cycles of length k in a random permutation is $1/k$.

(c) Deduce that the total number of cycles is $\ln n + \Theta(1)$.

Exercise 19 [Monte-Carlo Estimation] In *Monte-Carlo* estimation algorithms, the goal is to estimate the volume of a body by sampling. For instance, to estimate the value of π, we can randomly sample points in the $[-1, 1]^2$ square and count which ones are within the circle of radius 1 centred at the origin. Then we report the area to the be the count over four times the number of samples taken.

(a) Show that this algorithm produces the correct answer in expectation.

(b) What kind of guarantees can you get by applying Markov's inequality?

(c) What kind of guarantees can you get by applying Chebyshev's inequality?

(d) What kind of guarantees can you get by applying the Chernoff bound?

(e) How would you modify the algorithm so that you only sample from the square $[0, 1]^2$?

Exercise 20 [3-CNF] We are given a CNF formula consisting of m clauses, i.e., an expression of the form $C_1 \wedge C_2 \wedge \ldots \wedge C_m$ with $C_i = X_{i_1} \vee \ldots \vee X_{i_{c_i}}$. Consider a simple algorithm that randomly assigns a value to each of the literals X_i.

(a) Show that when $c_i = 3$ for all $i \in [m]$, then the output contains $\frac{7m}{8}$ satisfied clauses in expectation.

(b) Find a similar expression for the general case.

Exercise 21 [Pattern Matching] Consider a fixed pattern $P = (p_1, \ldots, p_k)$ with characters from an alphabet Σ. We want to find the expected number of times that this pattern occurs in a string X of length n whose characters are chosen uniformly at random from Σ. (For instance, the pattern "`abba`" appears three times in "`abaabbabbacabbacc`")

Exercise 22 [LCS Expectation] Consider two strings X and Y of length n whose characters are generated uniformly at random from the alphabet Σ. Show that the expected length of the LCS of these

two strings is $\Theta(n)$.

Exercise 23 [LIS Expectation Lower bound] In this exercise, you will prove that the length of the longest increasing subsequence in a random permutation of $[n] := \{1, \ldots, n\}$ is $\Omega(\sqrt{n})$.

(a) Consider any sequence X of length n. Further, let ℓ_i be the length of the LIS ending at character i and similarly u_i be the length of the longest decreasing subsequence ending at i.

i. Show that for any $i < j$ we must have that $(\ell_i, u_i) \neq (\ell_j, u_j)$.

ii. By considering the pairs with values (ℓ_i, u_i) such that $0 \leq \ell_i, u_i \leq \sqrt{n}/2$, argue that there must be an increasing or decreasing sequence with length at least $\sqrt{n}/2$.

(b) Deduce that in a random permutation, the length L of the longest common subsequence satisfies

$$\mathbf{E}[L] \geq \frac{\sqrt{n}}{4}.$$

1.4 Useful identities

Exercise 24 [Expectation as sum of probabilities]

(a) Consider a discrete random variable $X \geq 0$. Prove that

$$\mathbf{E}[X] = \sum_{i=1}^{\infty} \mathbf{Pr}[X \geq i].$$

(b) Use the above formula to derive the expectation of the geometric random variable $X \sim \mathsf{Geom}(p)$.

1.5 More inequalities

Exercise 25 [Reverse Markov's inequality] Consider a random variable X satisfying deterministically $X \leq b$ for some fixed c. Show that for any $a < \mathbf{E}[X]$.

$$\mathbf{Pr}[X > a] \geq \frac{\mathbf{E}[X] - a}{b - a}.$$

1.6 Coupon collector

Exercise 26 [Expectation] We consider the coupon collecting problem with n coupons. Prove that it takes $n \cdot \sum_{i=1}^{n} \frac{1}{i}$ days in expectation to collect all coupons.

Exercise 27 [High Probability] We consider the coupon collecting problem with n coupons. Deduce that the probability it takes more than $n \log n + cn$ days is at most e^{-c}.

2 Probabilistic Method

Exercise 28 Show that every graph $G = (V, E)$ contains a bipartite subgraph with at least $|E|/2$ edges.

3 Various Algorithms

Exercise 29 Design a randomised algorithm for the following problem. The input consists of an $n \times n$ matrix A with entries in $\{0,1\}$ and a vector x of length n with entries in the real interval $[0,1]$. The goal is to return a vector y of length n with entries in $\{0,1\}$ such that

$$\mathbf{Pr}\left[\max_{i=1,\ldots,n} |(Ax)_i - (Ay)_i| \leq 2\sqrt{n \log n}\right] \geq 1 - n^{-2}.$$

Hint: Your algorithm should have the property that $\mathbf{E}[A_{i,j} \cdot y_j] = \mathbf{E}[A_{i,j} \cdot x_j]$ for any $1 \leq i, j \leq n$.

Exercise 30 [Matrix Multiplication Verification] Given three $n \times n$ matrices A, B, C, we want to check if $A \times B = C$. The best known matrix multiplication algorithm takes time $\Omega(n^{2.37})$, so we would like to do the verification faster than that. For simplicity assume that the matrices are binary (i.e., their entries are just $0, 1$) and the operations are those modulo 2. Consider the algorithm that randomly samples a vector $v \in \{0,1\}^n$ and outputs 1 iff $(A \times B)v = Cv$.

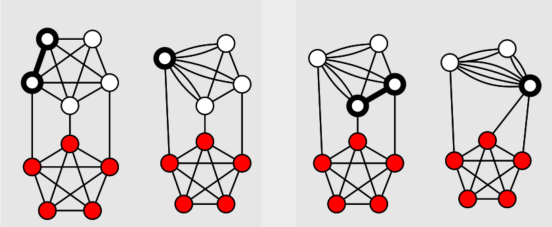
(a) Argue that the check $(A \times B)v = Cv$ can be performed more efficiently than matrix multiplication.

(b) Show that the algorithm succeeds with probability $\geq 1/2$.

(c) How would you improve the success probability?

(d) How would you apply the algorithm for other inputs?

Exercise 31 [Randomised Min-Cut] In this exercise, you will analyse a simple randomised algorithm for finding the minimum cut in an undirected graph. The algorithm proceeds by sampling a random edge (u, v) in the graph and *contracting* its endpoints. The figures below give two examples of contractions:



The algorithm makes $n - 2$ contractions until there are two vertices remaining. The number of edges between the two vertices will the answer for the minimum cut.

(a) Show that if the minimum cut in the graph has size M, then each vertex has degree at least M.

(b) Fix a minimum cut C. Let E_i be the event that in the i-th step the algorithm did not contract an edge of C. Show that

$$\mathbf{Pr}[E_1] \geq 1 - \frac{2}{n} \quad \text{and} \quad \mathbf{Pr}\left[E_i \;\middle|\; \bigcap_{j=1}^{i-1} E_j\right] \geq 1 - \frac{2}{n-i+1}.$$

(c) Prove that

$$\mathbf{Pr}\left[\bigcap_{j=1}^{n-2} E_j\right] \geq 1 - \frac{2}{n \cdot (n-1)}.$$

(d) Using amplification, design an algorithm with success probability at least $1 - n^{-1}$ of finding a minimum cut. What is the running time of your algorithm?

Exercise 32 [All min-cuts]

(a) Using Exercise 28 (c), argue that any graph can have at most $\binom{n}{2}$ min cuts.

(b) (optional) Show that there is a graph with this number of cuts.

(c) Modify the algorithm from Exercise 28 to return all minimum cuts for the given graph.

Exercise 33 [Karger-Stein algorithm] (++) Consider the following improvement to the algorithm in Exercise 28. For convenience we define the function `Contract`(G, t) to perform t random edge contractions on graph G.

```
1: function FasterMinCut(G = (V, E))
2:     if |V| ≤ 6 then
3:         return BruteForceMinCut(G)
4:     else
5:         t ← ⌈1 + |V|/√2⌉
6:         G_1 ← Contract(G, t)
7:         G_2 ← Contract(G, t)
8:     end if
9:     return min(FasterMinCut(G_1), FasterMinCut(G_2))
10: end function
```

(a) Show that the running time of `FasterMinCut`(G) for a graph with n vertices satisfies:

$$T(n) = 2T\left(\lceil 1 + n/\sqrt{2}\rceil\right) + \mathcal{O}(n^2).$$

(b) Show that $T(n) = \mathcal{O}(n^2 \log n)$.

(c) Show that a lower bound $P(n)$ on the probability of success for a graph with n vertices, satisfies the following recurrence

$$P(n) = 1 - \left(1 - \frac{1}{2}P\left(\lceil 1 + n/\sqrt{2}\rceil\right)\right).$$

(d) (++) Show that $P(n) = \Omega(\log n)$.

(e) Obtain an algorithm for min-cut with success probability at least $1 - n^{-1}$ and running time $\mathcal{O}(n \log^3 n)$.

Exercise 34 [Random subsets] You are given a finite set $S = \{s_1, \ldots, s_n\}$.

(a) Give an algorithm to generate a random subset of S.

(b) Let X and Y be two random subsets of S.

i. What is the probability that $X \subseteq Y$?

ii. What is the probability that $X \cup Y = S$?

iii. What is the expected size of $X \cup Y$?

iv. What is the expected size of $X \cap Y$?

Exercise 35 [Reservoir sampling] We are given a stream of N values, where N is large and unknown. The values will be presented one by one and we can only store one value at each step (perhaps because we have limited memory).
We want to sample from this stream one value uniformly at random.

(a) How would you sample the value if N was known?

(b) Consider an algorithm that stores the first value in memory and then for the k-th value a_k with probability $1/k$ it replaces the value in memory with a_k and otherwise it keeps it unchanged.

Show that this algorithm produces a uniform sample from the stream.

(c) Generalise the previous algorithm so that it samples M values uniformly at random from the stream. At each point in time your algorithm can keep at most M values in memory.

Exercise 36 [Algorithm L] Read about the Algorithm L for reservoir sampling and answer the following questions:

(a) Explain how this algorithm works (without having knowledge of the number of elements in the stream).

(b) In What sense is it better than the algorithm described in Exercise 32?

Exercise 37 [Algorithm A-Res] Read about Algorithm A-Res and answer the following questions:

(a) Describe the reservoir problem with weights.

(b) Explain how this algorithm works and prove its correctness.

Exercise 38 [Searching a Random Hash Table] In the **Lecture 2 slide 12**, it was shown that if we throw n balls into n bins, then the maximum load of any bin is at most $4 \cdot \log n / \log \log n$ with probability at least $1 - n^{-1}$.
In this exercise, we are interested in upper bounding the number of operations required for inserting n (random) elements in a hash table. The additional work is that when we insert an element v to a bin i with X_i elements, then we need to go over all of its elements to check if v is present. Therefore, this requires $\mathcal{O}(X_i)$ time.

(a) Using the analysis in the lecture, argue that the total time to insert n random keys to a hash table is $\mathcal{O}(n \cdot (\log n / \log \log n)^2)$ with high probability.

(b) Prove that for a binomial r.v. $N \sim \textsc{Bin}(n, p)$,

$$\mathbf{E}\left[\, N^2 \,\right] = n \cdot p + n \cdot (n-1) \cdot p^2.$$

(c) Using (b), show that the amortised insert time is $\mathcal{O}(n)$ in expectation.

4 Concentration inequalities

Exercise 39 [Chernoff Bound for Balls-into-Bins] Use the following Chernoff bound

$$\mathbf{Pr}\left[\, X \geq (1+\delta) \cdot \mathbf{E}\left[\, X \,\right] \,\right] \leq \exp\left(-\frac{\delta^2 \mathbf{E}\left[\, X \,\right]}{3}\right),$$

with an appropriate choice of δ to show that when allocating m balls into n bins uniformly at random, with $m \geq n \log n$, then the the maximum load of any bin is at most $\frac{m}{n} + \mathcal{O}(\sqrt{\frac{m}{n} \cdot \log n})$ with probability at least $1 - n^{-2}$.

Exercise 40 [Number of inversions Concentration] In Exercise 15, you proved that for a random permutation of elements $[n] := \{1, \ldots, n\}$, the number of inversions X is $\frac{1}{4} \cdot n(n-1)$ in expectation. In this exercise, you will apply McDiarmid's inequality to obtain a concentration bound.

(a) Argue that one can generate a random permutation of $[n]$ by sampling n independent values from $\mathcal{U}[0,1]$ and then replacing each with their rank (e.g., $(0.1, 0.41, 0.23, 0.21)$ would correspond to $(1, 4, 3, 2)$),

(b) Let f be the function that counts the number of inversions in a given permutation. Prove that f is Lipschitz with parameter n.

(c) Use McDiarmid's inequality to deduce that

$$\mathbf{Pr}\left[\, \left| X - \frac{1}{4} \cdot n(n-1) \right| < \sqrt{n^3 \log n} \,\right] \geq 1 - 2n^{-2}.$$

Exercise 41 [Number of Records Concentration] In Exercise 16, you proved that the expected number of records in a random permutation of elements in $[n]$ is $\log n + \Theta(1)$ in expectation. In this exercise, you will prove concentration.

(a) Let $\mathcal{E}_i$ be the event that element i is smaller than all previous elements. Then for any $i \neq j$ the events $\mathcal{E}_i$ and $\mathcal{E}_j$ are independent.

(b)

Exercise 42 [Local Maxima Concentration] In Exercise 17, you proved that the expected number of local maxima in a random permutation of $[n] := \{1, \ldots, n\}$ is $(n+1)/3$.
Prove that the number of local maxima is concentrated around this mean.

Exercise 43 [Longest Increasing Subsequence Concentration (I)] In this exercise, you will prove that the length of the longest increasing subsequence is $\mathcal{O}(\sqrt{n})$ with high probability.

(a) Let $X_{n,k}$ be the number of increasing subsquences of length k. Show that

$$\mathbf{E}[X_{n,k}] = \frac{1}{k!} \cdot \binom{n}{k}.$$

(b) Using Exercise 7, show that

$$\mathbf{E}[X_{n,k}] \leq \frac{n^k}{(k/e)^{2k}}.$$

(c) By arguing that

$$\mathbf{Pr}[L \geq k] \leq \mathbf{E}[X_{n,k}],$$

and using Markov's inequality, show that for any constant $C > 2$, we have that

$$\mathbf{Pr}\left[L \geq Ce\sqrt{n}\right] \leq \left(\frac{1}{C}\right)^{2Ce\sqrt{n}}.$$

(d) Deduce that $\mathbf{E}[L] = \mathcal{O}(\sqrt{n})$.

Exercise 44 [Longest Increasing Subsequence Concentration (II)] In Exercise 40, you proved that the length of the longest subsequence in a random permutation of $[n]$ is $\Theta(\sqrt{n})$ in expectation. As an alternative, use McDiarmid's inequality to prove that it is concentrated around the expectation.

Exercise 45 [Pattern Matching Concentration] In Exercise 21, we found the expectation for the number of occurrences N of pattern P (of length k) in a random string X (of length n).

(a) Prove that N is concentrated around its mean.

(b) For which values of N and k does your bound make sense?

Exercise 46 [Searching a Random Hash Table Concentration] In Exercise 35, you proved that the insertion time of n random elements in a hash table takes amortised $\mathcal{O}(n)$ time in expectation.
Let X_i be the loads of the bins after the n elements have been allocated, $Y_i = X_i^2$ and $Y_i' = \min(Y_i, \log^2 n)$. Further let $Y = \sum_{i=1}^{n} Y_i$ and $Y' = \sum_{i=1}^{n} Y_i'$.

(a) Using Exercise 35, argue that $\mathbf{E}[Y'] = \mathcal{O}(n)$.

(b) Argue that Y' is $(5 \log n)$-Lipschitz with respect to the n elements being inserted.

(c) Prove that Y' is concentrated around its mean.

(d) Deduce that Y is concentrated.

Exercise 47 [LCS Concentration] In Exercise 22, you proved that the length n of the longest common subsequence between two random strings X and Y is $\Theta(n)$.
Prove that L is concentrated around its mean.

Exercise 48 [Chromatic Number Concentration] The *chromatic number* $\chi(G)$ of a graph is the minimum number of colours to colour the vertices of G such that no two adjacent vertices have the same colour.
Consider an undirected graph G where each edge is inserted independently with probability p. Show that

$$\mathbf{Pr}\left[\,|\chi(G) - \mathbf{E}\left[\,\chi(G)\,\right]| \geq t\sqrt{n}\,\right] \leq 2 \cdot e^{-2t^2}.$$

Exercise 49 [Isolated vertices] Consider a graph sampled uniformly at random from the set of graphs with n vertices and cn edges for $c > 0$ being constant. Let X be the number of vertices that are *isolated*, i.e., have zero degree.

(a) Compute the expected value of X.

(b) Prove concentration for X.

Exercise 50 [Max-Cut Concentration] Consider an undirected, regular graph with degree d, i.e., every vertex has exactly d neighbours. Apply the McDiarmid's inequality in order to prove concentration for the randomised `Max-Cut` algorithm. What is the problem of applying it to an arbitrary graph?

5 Derandomisation

Exercise 51 [Derandomise Max-Cut] Derandomise the Max-cut algorithm that you saw in Lecture 1.

Exercise 52 [Derandomise 3-CNF] Derandomise the algorithm for the 3-CNF problem presented in Exercise 20.

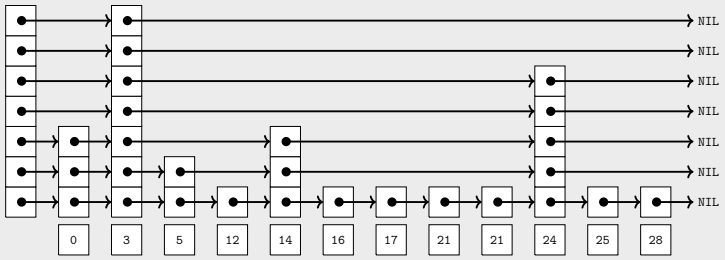
6 Randomised data structures

Exercise 53 [Randomised Binary Search Trees] In Part IA, you saw that Binary Search Trees without some balancing mechanism (e.g., rotations) can lead to depths that are $\Omega(n)$ in size.
In this exercise you will analyse the expected time complexity for a BST where the input values are random.

(a) What is the expected number of comparisons when building a RBST?

(b) What is the expected number of comparisons when searching for a random element in the RBST?

Exercise 54 [SkipList] The SkipList is a data structure like lists, but it aims to support $\mathcal{O}(\log n)$ access time to every element with high probability and it can also serve as a BST.
The idea is that when inserting an element x we also add a random number of pointers, generated by sampling a Geom(1/2) (the 1/2 not being important). For instance, here is a randomly generated skip list, where element 14 has 3 pointers. To search for an element we start from the top left and follow the pointers as long as we are before our target element.



(a) Show that for all of the n items have at most $\mathcal{O}(\log n)$ levels with probability at least $1 - n^{-2}$.

(b) By arguing backwards from the target element to the root, show that the search algorithm needs $\mathcal{O}(\log n)$ heads in $\Theta(\log n)$ coin flips and prove that this happens with high probability.

(c) (optional +) Argue that in total this data structure uses $\mathcal{O}(n)$ memory.

7 Chernoff Bounds

Exercise 55 [For Geometric r.vs.] Consider n independent geometric random variables $X_1, \ldots, X_n$ with $X_i \sim \textsc{Geom}(p)$.

(a) Prove that for any $t > 0$,

$$\mathbf{E}\left[\, e^{tX_i} \,\right] = \frac{p}{e^{-t} - 1 + p}.$$

(b) Using the inequality $1 + x \leq e^x$, show that

$$\mathbf{E}\left[\, e^{tX_i} \,\right] \leq \frac{p}{p - t} = \left(1 - \frac{t}{p}\right)^{-1}.$$

(c) Compute the Chernoff bound for $X := \sum_{i=1}^{n} X_i$ and optimise the choice of t to get for any $\delta > 0$,

$$\mathbf{Pr}\left[\, X \geq (1+\delta)\mathbf{E}\left[\, X \,\right] \,\right] \leq e^{-n \cdot (\delta - \ln(1+\delta))}.$$

8 Puzzles

8.1 Generating Random Variables

Exercise 56 You are given a biased coin which produces heads with probability p and tails with probability $1 - p$, for some unknown $p \in (0, 1)$.

- Design an algorithm that uses samples from this biased coin and produces an unbiased binary value.
- How efficient is your algorithm, i.e. how many biased samples does it need in expectation to generate an unbiased one (as a function of p)?

Exercise 57

- How would you use a (unbiased) random binary variables to generate random integers in the set $\{0, 1, \ldots, n\}$
- What is the expected running time of your algorithm?

Further Reading 3 You can find more problems of this sort here.

Exercise 58 [Random permutation] Design an algorithm that samples a permutation of n elements uniformly at random. How efficient is your algorithm?