1 Counting practice

You can leave your answers as (tidy) expressions involving factorials, binomial coefficients, etc., rather than evaluating them as decimal numbers.

1. How many different 13-card bridge hands contain exactly 4 spades? [Recall that there are 13 spades in a standard 52-card deck.A bridge hand is obtained by selecting 13 cards from a standard 52-card deck. The order of the cards within a bridge hand is considered irrelevant.]
2. How many ways are there to order a standard 52-card deck?
3. How many different anagrams of "ALABAMA" are there? (An anagram of "ALABAMA" is any re-ordering of the letters of "ALABAMA"; i.e., any string made up of the eight letters A, L, A, B, A, M, and A, in any order. The anagram does not have to be an English word.)
4. Suppose you are given 8 indistinguishable balls and 5 bins. How many distinguishable ways are there to distribute these 8 balls among these 5 bins such that no bin is empty? Assume the bins are distinguishable (e.g., numbered 1 through 5).
5. How many different ways are there to throw these 8 identical balls into 6 bins?

2 Hypercube routing

Recall that an n-dimensional hypercube contains 2^n vertices, each labeled with a distinct n bit string, and two vertices are adjacent iff their bit strings differ in exactly one position.

1. The hypercube is a popular architecture for parallel computation.Let each vertex of the hypercube represent a processor and each edge represent a communication link. Suppose we want to send a packet for vertex x to vertex y. Consider the following "bit-fixing" algorithm:

   > In each step, the current processor compares its address to the destination address of the packet. Let's say that the two addresses match up to the first k positions. The processor then forwards the packet and the destination address on to its neighboring processor whose address matches the destination address in at least the first $k+1$ positions. This process continues until the packet arrives at its destination.

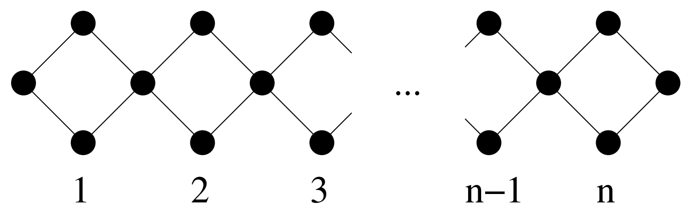
   Consider the following example where $n = 4$: Suppose that the source vertex is (1001) and the destination vertex is (0100). Give the sequence of processors that the packet is forwarded to using the bit-fixing algorithm.
2. In general, for an arbitrary source vertex and arbitrary destination vertex, how many edges must the packet traverse under this algorithm? Give an exact answer in terms of the n-bit strings labeling source and destination vertices. You *may* want to use the definition from **??**.

3. Consider any two vertices x and y in the hypercube. Consider the graph $G = (V_{x,y}, E_{x,y})$ where $E_{x,y}$ is the set of all edges on shortest paths between x and y and $V_{x,y}$ is the set of all vertices on shortest paths between x and y.

   Consider the following example where $n = 3$: Suppose that x is (010) and y is (100). What is the length of the shortest path between x and y? Explicitly show the sets $V_{x,y}$ and $E_{x,y}$.

3 Chains

Consider "chain" graphs, shaped like this (with n "links"):



How many different Eulerian tours, starting and ending at the leftmost vertex, does a chain graph with n links have? (Hint: Argue that there must be the first vertex after which an Eulerian tour takes an edge 'in the left direction'. There are 4 types of vertices in the graph, and only one them can be the vertex.

4 Error-correcting codes

In this question we will go through an example of error-correcting codes with general errors. We will send a message (m_0, m_1, m_2) of length $n = 3$. We will use an error-correcting code for $k = 1$ general error, doing arithmetic modulo 5.

(a) Suppose $(m_0, m_1, m_2) = (4, 3, 2)$. Use Lagrange interpolation to construct a polynomial $P(x)$ of degree 2 (remember all arithmetic is mod5) so that $(P(0), P(1), P(2)) = (m_0, m_1, m_2)$. Then extend the message to lengeth $n+2k$ by appending $P(3), P(4)$. What is the polynomial $P(x)$ and what is the message $(c_0, c_1, c_2, c_3, c_4) = (P(0), P(1), P(2), P(3), P(4))$ that is sent?

(b) Suppose the message is corrupted by changing c_0 to 0. We will locate the error using the Berlekamp–Welsh method. Let $E(x) = x + b_0$ be the error-locator polynomial, and $Q(x) = P(x)E(x) = a_3x^3 + a_2x^2 + a_1x + a_0$ be a polynomial with unknown coefficients. Write down the system of linear equations (involving unknowns a_0, a_1, a_2, a_3, b_0) in the Berlekamp–Welsh method. You need not solve the equations.

(c) The solution to the equations in part (b) is $b_0 = 0, a_0 = 0, a_1 = 4, a_2 = 4, a_3 = 0$. Show how the recipient can recover the original message (m_0, m_1, m_2).