CS 473 ✧ Fall 2022

Homework 0

Due Tuesday, August 30, 2022 at 9pm

- **This homework tests your familiarity with prerequisite material:** designing, describing, and analyzing elementary algorithms; fundamental graph problems and algorithms; and especially facility with recursion and induction. Notes on most of this prerequisite material are available on the course web page.
- **Each student must submit individual solutions for this homework.** For all future homeworks, groups of up to three students will be allowed to submit joint solutions.
- **Submit your solutions electronically on Gradescope as PDF files.**
  - Submit a separate PDF file for each numbered problem.
  - You can find a LaTeX solution template on the course web site; please use it if you plan to typeset your homework.
  - If you plan to submit scanned handwritten solutions, please use dark ink (not pencil) on blank white printer paper (not notebook or graph paper), and use a high-quality scanner or scanning app to create a high-quality PDF for submission (not a raw cell-phone photo). We reserve the right to reject submissions that are difficult to read.

☞ Some important course policies ☜

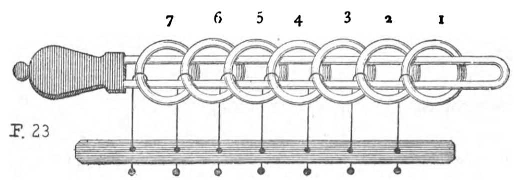
- **You may use any source at your disposal**—paper, electronic, or human—but you ***must*** cite ***every*** source that you use, and you ***must*** write everything yourself in your own words. See the academic integrity policies on the course web site for more details.
- **Avoid the Deadly Sins!** There are a few common writing (and thinking) practices that will be automatically penalized on every homework or exam problem. We're not just trying to be scary control freaks; history strongly suggests that people who commit these sins are more likely to make other serious mistakes as well. We're trying to break bad habits that seriously impede mastery of the course material.
  - Always give complete solutions, not just examples.
  - Every algorithm requires an English specification.
  - Never use weak induction. Weak induction should die in a fire.

See the course web site for more information.

If you have any questions about these policies,
please don't hesitate to ask in class, in office hours, or online.

---

1. The Tower of Hanoi is a relatively recent descendant of a much older mechanical puzzle known as the Baguenaudier, Chinese rings, Cardano's rings, Meleda, Patience, Tiring Irons, Prisoner's Lock, Spin-Out, and many other names. This puzzle was already well known in both China and Europe by the 16th century. The Italian mathematician Luca Pacioli described the 7-ring puzzle and its solution in his unpublished treatise *De Viribus Quantitatis*, written around 1500CE;[1] only a few years later, the Ming-dynasty poet Yang Shen described the 9-ring puzzle as "a toy for women and children".



A drawing of a 7-ring Baguenaudier, from *Récréations Mathématiques* by Édouard Lucas (1891)

The Baguenaudier puzzle has many physical forms, but it typically consists of a long metal loop and several rings, which are connected to a solid base by movable rods. The loop is initially threaded through the rings as shown in the figure above; the goal of the puzzle is to remove the loop.

More abstractly, we can model the puzzle as a sequence of bits, one for each ring, where the ith bit is 1 if the loop passes through the ith ring and 0 otherwise. Following tradition, we will index both the rings and the corresponding bits *from right to left*, as shown in the figure above. The puzzle allows two legal moves:

- Flip the rightmost bit.
- Flip the bit just to the left of the rightmost 1.

(The second move is impossible if the rightmost $n-1$ bits are all 0s.)

The goal of the puzzle is to transform a string of n 1s into a string of n 0s. For example, the following sequence of 21 moves solves the 5-ring puzzle:

$$\mathtt{11111} \xrightarrow{1} \mathtt{11110} \xrightarrow{3} \mathtt{11010} \xrightarrow{1} \mathtt{11011} \xrightarrow{2} \mathtt{11001} \xrightarrow{1} \mathtt{11000} \xrightarrow{5} \mathtt{01000}$$
$$\xrightarrow{1} \mathtt{01001} \xrightarrow{2} \mathtt{01011} \xrightarrow{1} \mathtt{01010} \xrightarrow{3} \mathtt{01110} \xrightarrow{1} \mathtt{01111} \xrightarrow{2} \mathtt{01101} \xrightarrow{1} \mathtt{01100} \xrightarrow{4} \mathtt{00100}$$
$$\xrightarrow{1} \mathtt{00101} \xrightarrow{2} \mathtt{00111} \xrightarrow{1} \mathtt{00110} \xrightarrow{3} \mathtt{00010} \xrightarrow{1} \mathtt{00011} \xrightarrow{2} \mathtt{00001} \xrightarrow{1} \mathtt{00000}$$

(a) Describe an algorithm to solve the Baguenaudier puzzle. Your input is the number of rings n; your algorithm should print a sequence of moves that solves the n-ring puzzle. For example, given the integer 5 as input, your algorithm should print the sequence $1, 3, 1, 2, 1, 5, 1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1$.

(b) *Exactly* how many moves does your algorithm perform, as a function of n? Prove your answer is correct.

(c) ***[Extra credit]*** Call a sequence of moves *reduced* if no move is the inverse of the previous move. Prove that for each non-negative integer n, there is *exactly one* reduced sequence of moves that solves the n-ring Baguenaudier puzzle.

[1] *De Viribus Quantitatis* [*On the Powers of Numbers*] is an important early work on recreational mathematics and perhaps the oldest surviving treatise on magic. Pacioli is better known for *Summa de Aritmetica*, a near-complete encyclopedia of late 15th-century mathematics, which included the first description of double-entry bookkeeping.

2. Prince Hutterdink and Princess Bumpercup are celebrating their recent marriage by inviting all the dukes and duchesses in the kingdom to sample the castle's celebrated wine cellars. Just before they drinking begins, the happy couple learns that exactly one of their n bottles of wine has been laced with iocaine powder, one of the deadliest poisons known to man.

   Hutterdink and Bumpercup employ several Royal Tasters who have built up an immunity to iocaine powder. (Strangely, every Royal Taster is named Roberts.) If a Taster consumes any amount of iocaine, no matter how small, they quickly become extremely ill, or as the Royal Miracle Workers optimistically put it, only *mostly* dead. Anyone else who consumes iocaine quickly becomes *all* dead.

   To test a set S of wine bottles, a Taster mixes one drop of wine from each bottle in S and consumes the resulting mixture. They will become mostly dead if and only if one of the bottles in S is poisoned. Each Taster must be paid 1000 guilders for each test they perform, so Hutterdink and Bumpercup want to use as few tests as possible. On the other hand, mostly dead Tasters require months to recover, and the party is tomorrow!

   (a) Suppose there is an unlimited supply of Tasters. Describe an algorithm to find the poisoned bottle using at most $O(\log n)$ tests. (This is best possible in the worst case.)
   (b) Now suppose there is only one Taster. Argue that $\Omega(n)$ tests are required in the worst case to find the poisoned bottle.
   (c) Now suppose there are two Tasters. Describe an algorithm that allows them to find the poisoned bottle using only $O(\sqrt{n})$ tests.
   (d) Finally, describe an algorithm to identify the poisoned bottle when there are k tasters. Report the number of tests that your algorithm uses as a function of both n and k.

3. At the start of the semester, the faculty and staff the See-Bull Center for Skeptical Media Consumption throw a welcome party for new and returning students. Every person who attended the party was given at least one rubber duck to keep in their office as a memento.

   When new graduate student Mariadne Inotaur arrives at the See-Bull Center the next day and asks for her rubber duck, she is told that all the rubber ducks are gone. Infuriated, Mariadne decides to return that night and steal as many ducks as she can.

   Some doors inside See-Bull are locked; each locked door requires a key card to open. There are six types of key cards, corresponding to different research groups that work in the building—Applesauce, Blarney, Claptrap, Drivel, Eyewash, and Flapdoodle. Each locked door can be unlocked by exactly one type of key card. Mariadne doesn't initially have any key cards, so if she wants to open a Flapdoodle door, for example, she must first find a Flapdoodle key card.

   Mariadne has a complete map of See-Bull, modeled as a sinple undirected graph $G = (V, E)$. The vertices V correspond to interior spaces (rooms, hallways, stairwells, and so on), plus one special vertex s for the outside world. Each vertex is labeled with the number of rubber ducks and the types of key cards (if any) in the corresponding space; there are no rubber ducks or key cards outside. The edges E correspond to doors between spaces. Each edge is labeled with a bit indicating whether the corresponding door is locked, and if so, the type of key card that unlocks it.

   Describe and analyze an algorithm to determine the maximum number of rubber ducks that Mariadne can steal, assuming she starts outside, with no rubber ducks or key cards.