

**Exercise 1.2** The minimal number of parameters (pairwise distances) that is required to completely specify a TSP instance is  $\binom{n}{2}$ , the binomial coefficient  $n$  over 2. Given a set with  $n$  elements it describes the number of subsets with exactly  $k$  (in our case 2) elements. (Subsets, thus the order is irrelevant, as required.)

**Exercise 1.4** The number of possible routes as a function of the number of cities is  $f(n) = [n]$ , the  $n$ th stirling number of the first kind with  $k = 1$ . There are  $n!$  different ways of arranging the  $n$  cities into a route, but many of them are identical from our perspective. See for example all  $4! = 24$  permutations of  $\{1, 2, 3, 4\}$ ,

$$\begin{aligned} (1, 2, 3, 4) &= (2, 3, 4, 1) = (3, 4, 1, 2) = (4, 1, 2, 3) \\ (2, 1, 3, 4) &= (3, 4, 2, 1) = (4, 2, 1, 3) = (1, 3, 4, 2) \\ (3, 1, 2, 4) &= (1, 2, 4, 3) = (2, 4, 3, 1) = (4, 3, 1, 2) \\ (1, 3, 2, 4) &= (4, 1, 3, 2) = (2, 4, 1, 3) = (3, 2, 4, 1) \\ (2, 3, 1, 4) &= (4, 2, 3, 1) = (3, 1, 4, 2) = (1, 4, 2, 3) \\ (3, 2, 1, 4) &= (4, 3, 2, 1) = (1, 4, 3, 2) = (2, 1, 4, 3) \end{aligned}$$

which only admit six different *cycles* of length four. The expression  $[n]$  represents the number of permutations over  $n$  elements with one cycle, which is what we are looking for.

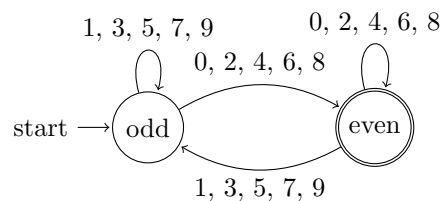
**Problem 1.9** See Exercises 1.2 and 1.4.

**Numericals 1.10** Not implemented, getting the pairwise distances would be a hassle.

**Problem 1.11** We want to show that: „Every symbol from a given alphabet can be represented as a bitstring. This representation is one-to-one and only scales logarithmically in alphabet size.“ Consider an alphabet  $A$  with  $n$  symbols where each symbol  $a_0, a_1, \dots, a_{n-1}$  can be represented by an unique integer. Because we can use the index of an element as its integer counterpart, at least one such representation exists

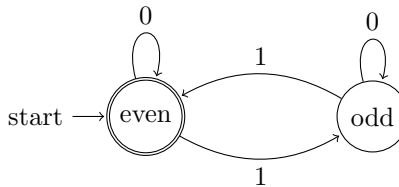
We can uniquely represent an integer  $c$  as a bitstring  $b$  through the following algorithm: Zero all bits in  $b$ . Find the largest  $x$  such that  $2^x$  divides  $c$ . Set  $b_x$  to one and set  $c$  to  $c - 2^x$ . Continue until  $c$  is zero. The length of such a bitstring is determined by the  $x$  of the first iteration, so the first  $x$  such that  $2^x \mid c$ . Given only  $c$ , the length of the resulting bitstring will thus be  $\lfloor \log_2(c) \rfloor$ .

**Exercise 2.1** The relevant state diagram is



The runtime of this machine is equal to the length of the decimal encoding, thus  $\lfloor \log_{10}(n) \rfloor + 1$ . This is very similar to the runtime of a binary parity checking machine since both of them have logarithmic growth, but because  $\log_{k+1}$  grows slower than  $\log_k$  our machine is slightly faster.

**Exercise 2.2** The relevant state diagram is



For  $n = 2$  it is equivalent to XNOR.

**Exercise 2.4** No.

**Exercise 2.6** No.