of algorithm design. Finally, the book of Motwani and Raghavan[MR95] is an excellent survey of the field of randomized algorithms.

The modern theory of computational complexity was especially influenced by the papers of Cook[Coo71] and Karp[Kar72]. Many similar ideas were arrived at independently in Russia by Levin[Lev73], but unfortunately took time to propagate to the West. The classic book by Garey and Johnson[GJ79] has also had an enormous influence on the field. More recently, Papadimitriou[Pap94] has written a beautiful book that surveys many of the main ideas of computational complexity theory. Much of the material in this chapter is based upon Papadimitriou's book. In this chapter we considered only one type of reducibility between languages, polynomial time reducibility. There are many other notions of reductions between languages. An early survey of these notions was given by Ladner, Lynch and Selman[LLS75]. The study of different notions of reducibility later blossomed into a subfield of research known as *structural complexity*, which has been reviewed by Balcázar, Diaz, and Gabarró[BDG88a, BDG88b].

The connection between information, energy dissipation, and computation has a long history. The modern understanding is due to a 1961 paper by Landauer[Lan61], in which Landauer's principle was first formulated. A paper by Szilard[Szi29] and a 1949 lecture by von Neumann[von66] (page 66) arrive at conclusions close to Landauer's principle, but do not fully grasp the essential point that it is the *erasure* of information that requires dissipation.

Reversible Turing machines were invented by Lecerf[Lec63] and later, but independently, in an influential paper by Bennett[Ben73]. Fredkin and Toffoli[FT82] introduced reversible circuit models of computation. Two interesting historical documents are Barton's May, 1978 MIT 6.895 term paper[Bar78], and Ressler's 1981 Master's thesis[Res81], which contain designs for a reversible PDP-10! Today, reversible logic is potentially important in implementations of low-power CMOS circuitry[YK95].

Maxwell's demon is a fascinating subject, with a long and intricate history. Maxwell proposed his demon in 1871[Max71]. Szilard published a key paper in 1929[Szi29] which anticipated many of the details of the final resolution of the problem of Maxwell's demon. In 1965 Feynman[FLS65b] resolved a special case of Maxwell's demon. Bennett, building on Landauer's work[Lan61], wrote two beautiful papers on the subject[Ben82, Ben87] which completed the resolution of the problem. An interesting book about the history of Maxwell's demon and its exorcism is the collection of papers by Leff and Rex[LR90].

DNA computing was invented by Adleman, and the solution of the directed Hamiltonian path problem we describe is his[Adl94]. Lipton has also shown how 3SAT and circuit satisfiability can be solved in this model[Lip95]. A good general article is Adleman's *Scientific American* article[Adl98]; for an insightful look into the universality of DNA operations, see Winfree[Win98]. An interesting place to read about performing reliable computation in the presence of noise is the book by Winograd and Cowan[WC67]. This topic will be addressed again in Chapter 10. A good textbook on computer architecture is by Hennessey, Goldberg, and Patterson.[HGP96].

Problems 3.1 through 3.4 explore a line of thought originated by Minsky (in his beautiful book on computational machines[Min67]) and developed by Conway[Con72, Con86]. The Fractran programming language is certainly one of the most beautiful and elegant universal computational models known, as demonstrated by the following example, known