II Quantum computation

4 Quantum circuits

The theory of computation has traditionally been studied almost entirely in the abstract, as a topic in pure mathematics. This is to miss the point of it. Computers are physical objects, and computations are physical processes. What computers can or cannot compute is determined by the laws of physics alone, and not by pure mathematics.
– David Deutsch

Like mathematics, computer science will be somewhat different from the other sciences, in that it deals with artificial laws that can be proved, instead of natural laws that are never known with certainty.
– Donald Knuth

The opposite of a profound truth may well be another profound truth.
– Niels Bohr

This chapter begins Part II of the book, in which we explore quantum computation in detail. The chapter develops the fundamental principles of quantum computation, and establishes the basic building blocks for quantum circuits, a universal language for describing sophisticated quantum computations. The two fundamental quantum algorithms known to date are constructed from these circuits in the following two chapters. Chapter 5 presents the quantum Fourier transform and its applications to phase estimation, order-finding and factoring. Chapter 6 describes the quantum search algorithm, and its applications to database search, counting and speedup of solutions to **NP**-complete problems. Chapter 7 concludes Part II with a discussion of how quantum computation may one day be experimentally realized. Two other topics of great interest for quantum computation, quantum noise and quantum error-correction, are deferred until Part III of the book, in view of their wide interest also *outside* quantum computation.

There are two main ideas introduced in this chapter. First, we explain in detail the fundamental model of quantum computation, the quantum circuit model. Second, we demonstrate that there exists a small set of gates which are *universal*, that is, any quantum computation whatsoever can be expressed in terms of those gates. Along the way we also have occasion to describe many other basic results of quantum computation. Section 4.1 begins the chapter with an overview of quantum algorithms, focusing on what algorithms are known, and the unifying techniques underlying their construction. Section 4.2 is a detailed study of single qubit operations. Despite their simplicity, single qubit operations offer a rich playground for the construction of examples and techniques, and it is essential to understand them in detail. Section 4.3 shows how to perform multi-qubit *controlled unitary* operations, and Section 4.4 discusses the description of measurement in the quantum circuits model. These elements are then brought together in Section 4.5 for the statement and proof of the universality theorem. We summarize all the basic elements

of quantum computation in Section 4.6, and discuss possible variants of the model, and the important question of the relationship in computational power between classical and quantum computers. Section 4.7 concludes the chapter with an important and instructive application of quantum computation to the *simulation* of real quantum systems.

This chapter is perhaps the most reader-intensive of all the chapters in the book, with a high density of exercises for you to complete, and it is worth explaining the reason for this intensity. Obtaining facility with the basic elements of the quantum circuit model of computation is quite easy, but requires assimilating a large number of simple results and techniques that must become second nature if one is to progress to the more difficult problem of designing quantum algorithms. For this reason we take an example-oriented approach in this chapter, and ask you to fill in many of the details, in order to acquire such a facility. A less intensive, but somewhat superficial overview of the basic elements of quantum computation may be obtained by skipping to Section 4.6.

4.1 Quantum algorithms

What is a quantum computer good for? We're all familiar with the frustration of needing more computer resources to solve a computational problem. Practically speaking, many interesting problems are impossible to solve on a classical computer, not because they are in principle insoluble, but because of the astronomical resources required to solve realistic cases of the problem.

The spectacular promise of quantum computers is to enable new algorithms which render feasible problems requiring exorbitant resources for their solution on a classical computer. At the time of writing, two broad classes of quantum algorithms are known which fulfill this promise. The first class of algorithms is based upon Shor's *quantum Fourier transform*, and includes remarkable algorithms for solving the factoring and discrete logarithm problems, providing a striking *exponential* speedup over the best known classical algorithms. The second class of algorithms is based upon Grover's algorithm for performing *quantum searching*. These provide a less striking but still remarkable *quadratic* speedup over the best possible classical algorithms. The quantum searching algorithm derives its importance from the widespread use of search-based techniques in classical algorithms, which in many instances allows a straightforward adaptation of the classical algorithm to give a faster quantum algorithm.

Figure 4.1 sketches the state of knowledge about quantum algorithms at the time of writing, including some sample applications of those algorithms. Naturally, at the core of the diagram are the quantum Fourier transform and the quantum searching algorithm. Of particular interest in the figure is the quantum counting algorithm. This algorithm is a clever combination of the quantum searching and Fourier transform algorithms, which can be used to estimate the number of solutions to a search problem more quickly than is possible on a classical computer.

The quantum searching algorithm has many potential applications, of which but a few are illustrated. It can be used to extract statistics, such as the minimal element, from an unordered data set, more quickly than is possible on a classical computer. It can be used to speed up algorithms for some problems in **NP** – specifically, those problems for which a straightforward search for a solution is the best algorithm known. Finally, it can be used to speed up the search for keys to cryptosystems such as the widely used Data Encryption Standard (DES). These and other applications are explained in Chapter 6.

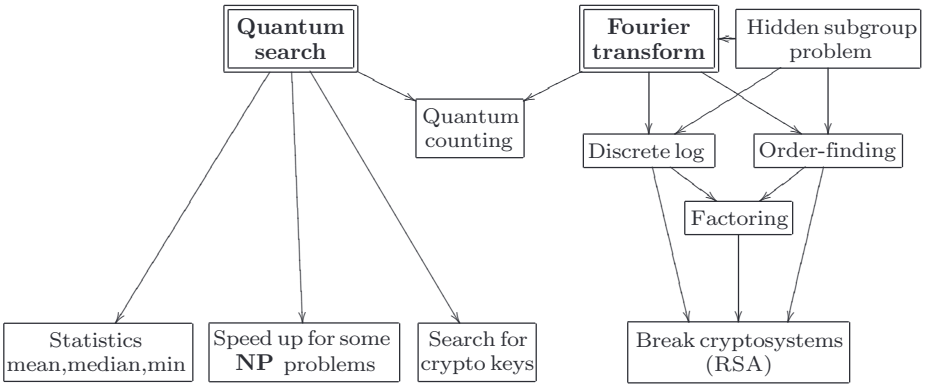

Figure 4.1. The main quantum algorithms and their relationships, including some notable applications.

The quantum Fourier transform also has many interesting applications. It can be used to solve the discrete logarithm and factoring problems. These results, in turn, enable a quantum computer to break many of the most popular cryptosystems now in use, including the RSA cryptosystem. The Fourier transform also turns out to be closely related to an important problem in mathematics, finding a hidden subgroup (a generalization of finding the period of a periodic function). The quantum Fourier transform and several of its applications, including fast quantum algorithms for factoring and discrete logarithm, are explained in Chapter 5.

Why are there so few quantum algorithms known which are better than their classical counterparts? The answer is that coming up with good quantum algorithms seems to be a difficult problem. There are at least two reasons for this. First, algorithm design, be it classical or quantum, is not an easy business! The history of algorithms shows us that considerable ingenuity is often required to come up with near optimal algorithms, even for apparently very simple problems, like the multiplication of two numbers. Finding good quantum algorithms is made doubly difficult because of the additional constraint that we want our quantum algorithms to be *better* than the best known classical algorithms. A second reason for the difficulty of finding good quantum algorithms is that our intuitions are much better adapted to the classical world than they are to the quantum world. If we think about problems using our native intuition, then the algorithms which we come up with are going to be classical algorithms. It takes special insights and special tricks to come up with good quantum algorithms.

Further study of quantum algorithms will be postponed until the next chapter. In this chapter we provide an efficient and powerful language for describing quantum algorithms, the language of quantum circuits – assemblies of discrete sets of components which describe computational procedures. This construction will enable us to quantify the cost of an algorithm in terms of things like the total number of gates required, or the circuit depth. The circuit language also comes with a toolbox of tricks that simplifies algorithm design and provides ready conceptual understanding.