

Problem set 2

Problem and exercise numbers are from CLRS 2nd edition.

Remember that to *give an algorithm* means not only to describe the algorithm, but also to analyze its running time.

- Where in the matrix multiplication-based DP algorithm for the all-pairs shortest paths problem do we need the associativity of matrix multiplication?
- Let $G = (V, E, w)$ be a weighted graph. A *cycle* in G is a sequence $C = (e_0, e_1, \dots, e_k)$ of edges such that e_i, e_{i+1} are adjacent for $i = 0 \dots k-1$ and e_k is adjacent to e_0 . Let $w(C)$ denote the total weight of edges in C . Assume that G has no negative cycles; i.e., for any cycle C , $w(C) \geq 0$. Give an algorithm to find a cycle C that minimizes $\frac{w(C)}{\sqrt{|C|}}$, where $|C| = 1(C)$ is the number of edges in C .
- 15-1 Bitonic euclidean traveling-salesman problem. The euclidean traveling-salesman problem is the problem of determining the shortest closed tour that connects a given set of n points in the plane. Figure 15.9(a) shows the solution to a 7-point problem. The general problem is NP-complete, and its solution is therefore believed to require more than polynomial time (see Chapter 34).

J. L. Bentley has suggested that we simplify the problem by restricting our attention to bitonic tours, that is, tours that start at the leftmost point, go strictly left to right to the rightmost point, and then go strictly right to left back to the starting point. Figure 15.9(b) shows the shortest bitonic tour of the same 7 points. In this case, a polynomial-time algorithm is possible.

Describe an $O(n^2)$ -time algorithm for determining an optimal bitonic tour. You may assume that no two points have the same x -coordinate. (Hint: Scan left to right, maintaining optimal possibilities for the two parts of the tour.)

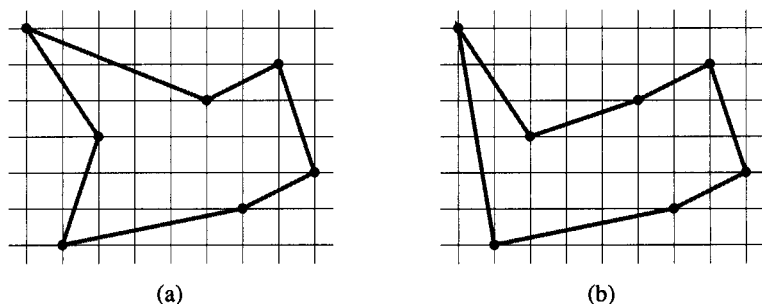


Figure 15.9 Seven points in the plane, shown on a unit grid. (a) The shortest closed tour, with length approximately 24.89. This tour is not bitonic. (b) The shortest bitonic tour for the same set of points. Its length is approximately 25.58.

- 15-2 Printing neatly. Consider the problem of neatly printing a paragraph on a printer. The input text is a sequence of n words of lengths l_1, l_2, \dots, l_n , measured in characters. We want to print this paragraph neatly on a number of lines that hold a maximum of M characters each. Our criterion of "neatness" is as follows. If a given line contains words i through j , where $i < j$, and we leave exactly one space between words, the number of extra space characters at the end of the line is $M - j + i - \sum_{k=i}^j l_k$, which must be nonnegative so that the words fit on the line. We wish to minimize the sum, over all

lines except the last, of the cubes of the numbers of extra space characters at the ends of lines. Give a dynamic-programming algorithm to print a paragraph of n words neatly on a printer. Analyze the running time and space requirements of your algorithm.

- 25.1-9 Modify Faster-All-Pairs-Shortest-Paths so that it can detect the presence of a negative-weight cycle.

FASTER-ALL-PAIRS-SHORTEST-PATHS(W)

```

1   $n \leftarrow \text{rows}[W]$ 
2   $L^{(1)} \leftarrow W$ 
3   $m \leftarrow 1$ 
4  while  $m < n - 1$ 
5      do  $L^{(2m)} \leftarrow \text{EXTEND-SHORTEST-PATHS}(L^{(m)}, L^{(m)})$ 
6           $m \leftarrow 2m$ 
7  return  $L^{(m)}$ 
```

- 25.1-10 Give an efficient algorithm to find the length (number of edges) of a minimum-length negative-weight cycle in a graph.