作业 3：

（双调欧几里得旅行商问题） 在**欧几里得旅行商问题**中，给定平面上 n 个点作为输入，希望求出连接所有 n 个点的最短巡游路线。图 15-11(a)给出了一个 7 点问题的解。此问题是 NP 难问题，因此大家相信它并不存在多项式时间的求解算法(参见第 34 章)。

J. L. Bentley 建议将问题简化，限制巡游路线为**双调巡游**(bitonic tours)，即从最左边的点开始，严格向右前进，直至最右边的点，然后调头严格向左前进，直至回到起始点。图 15-11(b)给出了相同 7 个点的最短双调巡游路线。问题简化之后，存在一个多项式时间的算法。

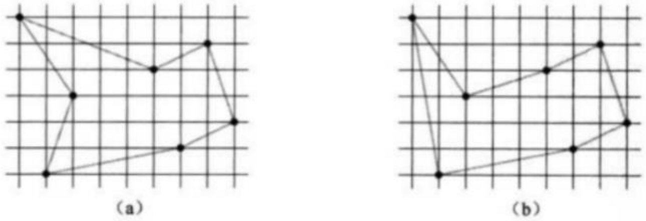

图 15-11 给定平面上 7 个点，显示在一个单位网格中。(a)最短闭合巡游路线，长度约为 24.89，此路线不是双调的。(b)同样点集的最短双调巡游路线，长度约为 25.58

设计一个 $O(n^2)$时间的最优双调巡游路线算法。你可以认为任何两个点的 x 坐标均不同，且所有实数运算都花费单位时间。(提示：由左至右扫描，对巡游路线的两个部分分别维护可能的最优解。)

题目要求：设计 O（n^2）的动态规划算法。

（1）证明最优子结构

（2）写出递归关系式

（3）写出伪代码

No.

Date

2018330500271-叶城宇 -7

解：对n个点按x坐标从小到大排序$P_1, P_2, \cdots, P_n$，其中P_1为最左边的点，P_n为最右边的点。假设P_{ij} $(i<j)$表示一条包含$P_1, P_2, \cdots P_n$的最短双调路径，这条路从P_i向左直到P_1，然后从P_1向右直到P_j。显然，$P_{n,n}$就是需求的路径，P_i与P_j间距为：

$$\|P_iP_j\| = \sqrt{(x_i-x_j)^2 + (y_i-y_j)^2}$$

(1) 当 $i<j-1$ 时

$$L(i,j) = L(i,j-1) + \|P_{j-1}P_j\|$$

(2) 当 $i=j-1$ 时

$$L(i,j) = \min_{1\le k\le j-2} (L(k,j-1) + \|P_kP_j\|$$

(3) 当 $i=j$ 时.

$$L(n,n) = L(n-1,n) + \|P_{n-1}P_n\|$$

递归关系式.

伪代码：MIN-BITONIC-PATH(P)

```
n = P.length
creat 2 new 2-D arrays L[1..n, 2..n] and left [1...n, 2..n]
L(1,2) = ||P1 P2||
Left(1,2) = P1
 for j = 3 to n
     for i = 1 to j-2
         L(i,j) = L(i,j-1) + ||Pj-1 Pj||
         left(i,j) = Pj-1
```

```
L(j-1, j) = ∞
for k = 1 to j-2
    q = L(k, j-1) + ||pk pj||
    if q < L(j-1, j)
        L(j-1, j) = q
        Left(j-1, j) = pk
L(n,n) = L(n-1, n) + ||Pn-1 Pn||
Left(n,n) = Pn-1
return L and Left.

PRINT-MIN-BITONIC-PATH(left, i, j)
  if i >= j
    if i == 2 && j = 1
      return
    print left(i, j)
    PRINT-MIN-BITONIC-PATH(left, left(j, i), j)
  else
    if i == 1 && j == 2
      return
    PRINT-MIN-BITONIC-PATH(left, i, left(i, j))
    print Left(i, j)
```