CHAPTER 7

Permutation Graphs

1. Introduction

In this chapter we consider a class of perfect graphs which has a large number of applications. Suppose π is a permutation of the numbers $1, 2, \ldots, n$. Let us think of π as the sequence $[\pi_1, \pi_2, \ldots, \pi_n]$, so, for example, the permutation $\pi = [4, 3, 6, 1, 5, 2]$ has $\pi_1 = 4$, $\pi_2 = 3$, etc. Notice that $(\pi^{-1})_i$, denoted here as π_i^{-1}, is the *position in the sequence* where the number i can be found; in our example $\pi_4^{-1} = 1$, $\pi_3^{-1} = 2$, etc.

We can construct an undirected graph $G[\pi]$ from π in the following manner: $G[\pi]$ has vertices numbered from 1 to n; two vertices are joined by an edge if the larger of their corresponding numbers is to the left of the smaller in π (that is, they occur out of their proper order reading left to right). In our example, both 4 and 3 are connected to 1 since they are each larger and to the left of 1, whereas neither 5 nor 2 is connected to 1 (see Figure 7.1). The graph $G[\pi]$ is sometimes called the *inversion graph* of π.

More formally, if π is a permutation of the numbers $1, 2, \ldots, n$, then the graph $G[\pi] = (V, E)$ is defined as follows:

$$V = \{1, 2, \ldots, n\}$$

and

$$ij \in E \Leftrightarrow (i - j)(\pi_i^{-1} - \pi_j^{-1}) < 0.$$

An undirected graph G is called a *permutation graph* if there exists a permutation π such that $G \cong G[\pi]$.

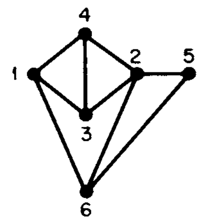


Figure 7.1. The graph $G[4, 3, 6, 1, 5, 2]$.

2. Characterizing Permutation Graphs

Permutation graphs have many interesting properties. Notice what happens when we *reverse* the sequence π. Each pair of numbers which occurred in the correct order in π is now in the wrong order, and vice versa. Thus, the permutation graph we obtain is the complement of $G[\pi]$. In other words, if π^ρ is the permutation obtained by reversing the sequence π, then

$$G[\pi^\rho] = \overline{G[\pi]}.$$

This shows that *the complement of a permutation graph is also a permutation graph.*

Another property of the graph $G[\pi]$ (which you may have already guessed) is that it is transitively orientable. If we orient each edge toward its larger endpoint, then we will obtain a transitive orientation F. For, suppose $ij \in F$ and $jk \in F$, then $i < j < k$ and $\pi_i^{-1} > \pi_j^{-1} > \pi_k^{-1}$, which implies that $ik \in F$. This result is only half of the story; we actually have the following:

Theorem 7.1 (Pnueli, Lempel, and Even [1971]). An undirected graph G is a permutation graph if and only if G and $\bar{G}$ are comparability graphs.

Proof. Suppose $G \cong G[\pi]$; then G is a comparability graph since $G[\pi]$ has a transitive orientation. Likewise, $\bar{G}$ is a comparability graph since $\bar{G} \cong G[\pi^\rho]$.

Conversely, let (V, F_1) and (V, F_2) be transitive orientations of $G = (V, E)$ and $\bar{G} = (V, \bar{E})$, respectively. We claim that $(V, F_1 + F_2)$ is an *acyclic* orientation of the complete graph $(V, E + \bar{E})$. For suppose $F_1 + F_2$ had a cycle $[v_0, v_1, v_2, \ldots, v_l, v_0]$ of the smallest possible length l. If $l > 3$, then the cycle can be shortened either by $v_0 v_2$ or $v_2 v_0$, contradicting minimality. If $l = 3$, then at least two of the edges of the cycle are in the same F_i, implying that F_i is not transitive. Thus $(V, F_1 + F_2)$ is acyclic. Similarly $(V, F_1^{-1} + F_2)$ is acyclic.

We conclude the proof by constructing a permutation π such that $G \cong G[\pi]$. An acyclic orientation of a complete graph is transitive, and it determines a unique linear ordering of the vertices. (See Section 2.5 on transitive tournaments.) Consider the following procedure.

Step I. Label the vertices according to the order determined by $F_1 + F_2$; namely, the vertex x of in-degree $i - 1$ gets label $L(x) = i$.

Step II. Label the vertices according to the order determined by $F_1^{-1} + F_2$; namely, the vertex x of in-degree $i - 1$ gets label $L'(x) = i$.

Notice that

$$xy \in E \Leftrightarrow [L(x) - L(y)][L'(x) - L'(y)] < 0, \tag{1}$$

since it is the edges of E which have their orientations reversed between steps I and II. This is the key to our argument.

Step III. Define π as follows: For each vertex x, if $L(x) = i$, then $\pi_i^{-1} = L'(x)$. The relationship is depicted in the commuting diagram below.

$$\begin{tikzcd} V \arrow[d, "L'"'] \arrow[dr, "L"] & \\ \{1, 2, \ldots, n\} \arrow[r, "\pi"] & \{1, 2, \ldots, n\} \end{tikzcd}$$

Therefore, by (1), π is the desired permutation and L is the desired isomorphism. ∎

Remark. In terms of the nomenclature of Section 5.8, G is a permutation graph if and only if the transitive orientations of G, when regarded as partial orders have dimension at most 2.

The construction technique presented above is illustrated in Figure 7.2.

Theorem 7.1 suggests an algorithm for recognizing permutation graphs, namely, applying the transitive orientation algorithm to the graph and to its complement. If we succeed in finding transitive orientations, then the graph is a permutation graph. To find a suitable permutation we can follow the construction procedure in the proof of the theorem. The entire method requires $O(n^3)$ time and $O(n^2)$ space for a graph with n vertices.

We conclude this section with a remark which follows from transitive orientability.

Remark. The decreasing subsequences of π and the cliques of $G[\pi]$ are in one-to-one correspondence. The increasing subsequences of π and the stable sets of $G[\pi]$ are in one-to-one correspondence.

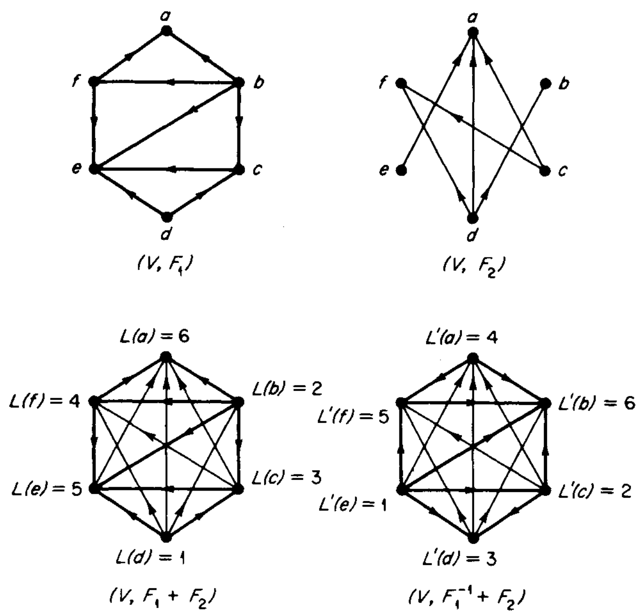


Figure 7.2. Construction of the permutation $\pi = [5, 3, 1, 6, 4, 2]$ from the transitive orientations F_1 and F_2. Vertex a gives $\pi_6^{-1} = 4$, vertex b gives $\pi_2^{-1} = 6$, etc.

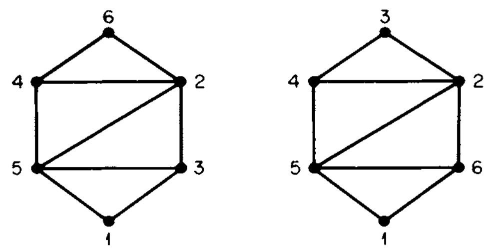
3. Permutation Labelings

A related, but simpler, problem is that of testing whether a *given* labeling of the vertices of a graph is a permutation labeling. Let $G = (V, E)$ be an undirected graph, and let $L: V \to \{1, 2, \ldots, n\}$ be a bijection labeling the vertices. We call L a *permutation labeling* if there exists a permutation π of $\{1, 2, \ldots, n\}$ such that

$$xy \in E \Leftrightarrow [L(x) - L(y)][\pi^{-1}(L(x)) - \pi^{-1}(L(y))] < 0.$$

Clearly, G is a permutation graph if and only if it has at least one permutation labeling.

Figure 7.3 shows two labelings of the same graph. The first is the permutation labeling already constructed in Figure 7.2. The second is not a permutation labeling for the following reason. Since $\mathrm{Adj}(1) = \{5, 6\}$, both 5 and 6 would be on the left of 1 while 2–4 would be on the right of 1 in any permutation π that might work. However, this implies that 3 and 4 would be to the right of 6—yet they are not connected to 6. Hence, no such permutation π exists for this labeling.



This labeling is a permutation labeling

This labeling is not a permutation labeling

Figure 7.3.

Theorem 7.2 (Gill and Acharya [1977]). Let $G = (V, E)$ be an undirected graph. A bijection $L: V \to \{1, 2, \ldots, n\}$ is a permutation labeling of G if and only if the mapping

$$F: x \to L(x) - d^-(x) + d^+(x) \qquad (x \in V)$$

is an injection, where

$$d^-(x) = |\{y \in \operatorname{Adj}(x) \mid L(y) < L(x)\}|$$

and

$$d^+(x) = |\{y \in \operatorname{Adj}(x) \mid L(y) > L(x)\}|.$$

Proof. ($\Rightarrow$) Let π be a permutation corresponding to the labeling L. Then $d^-(x)$ is the number of integers in π smaller than and to the right of $L(x)$, and $d^+(x)$ is the number of integers in π larger than and to the left of $L(x)$. By Exercise 4, $f(x) = \pi^{-1}(L(x))$, and since π^{-1} and L are injective (indeed bijective), so too is f.

($\Leftarrow$) Assuming that f is injective, we will construct the desired permutation. Since $d^-(x) \leq L(x) - 1$ and $d^+(x) \leq n - L(x)$, it follows that

$$1 \leq f(x) \leq n \qquad (x \in V). \tag{2}$$

But f is injective and integer valued, so (2) implies that f is a bijection from V to $\{1, 2, \ldots, n\}$. Define π as follows:

$$\pi(i) = L(f^{-1}(i))$$

(see Figure 7.4).

$$\begin{tikzcd} V \arrow[d, "f"'] \arrow[dr, "L"] & \\ \{1, 2, \ldots, n\} \arrow[r, "\pi"] & \{1, 2, \ldots, n\} \end{tikzcd}$$

Figure 7.4.

Now, π is a permutation, since L and f^{-1} are bijective. Furthermore,

$$\pi^{-1}(L(x)) = f(x) \qquad (x \in V),$$

so we must verify that

$$xy \in E \Leftrightarrow [L(x) - L(y)][f(x) - f(y)] < 0.$$

This is left as an exercise for the reader. ∎

4. Applications

Permutation graphs can be regarded as a class of intersection graphs in the following manner. Write the numbers $1, 2, \ldots, n$ horizontally from left to right; underneath them write the numbers $\pi_1, \pi_2, \ldots, \pi_n$ in sequence, again horizontally left to right; finally, draw n straight line segments joining the two 1's, the two 2's, etc. We call this the *matching diagram* of π (see Figure 7.5). Notice that the ith segment intersects the jth segment if and only if i and j appear in reversed order in π; this is the same criterion for the vertices i and j of $G[\pi]$ to be adjacent. Therefore, the intersection graphs of the segments of matching diagrams are exactly the permutation graphs.

The reason for our introducing these matching diagrams is to assist us in studying some applications of permutation graphs.

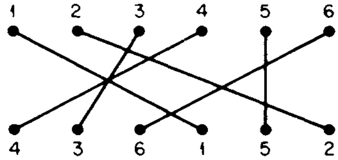


Figure 7.5. The matching diagram of [4, 3, 6, 1, 5, 2].

Application 7.1. Suppose we have two collections of cities, the X cities and the Y cities, lying, respectively, on two parallel lines. Suppose also, that there are airline routes connecting various X cities with various Y cities, all scheduled to be utilized at the same time of the day. Our mission, should we decide to accept it, will be to assign altitudes to each flight path so that intersecting routes will be at different altitudes. We will thereby assure that no midair collisions will occur. Being clever graph theorists, we recognize this as a coloring problem.

The data, as given, provides us with a bipartite graph embedded in the plane, as pictured in Figure 7.6. We number the flight paths by traversing the northern cities from west to east. From this we can extract a matching diagram, or go straight to the corresponding permutation graph $G[\pi]$. Assigning altitudes to the flight paths so that intersecting paths receive different altitudes is equivalent to coloring the vertices of $G[\pi]$ so that adjacent vertices receive different colors. An efficient coloring algorithm for permutation graphs is given in the next section.

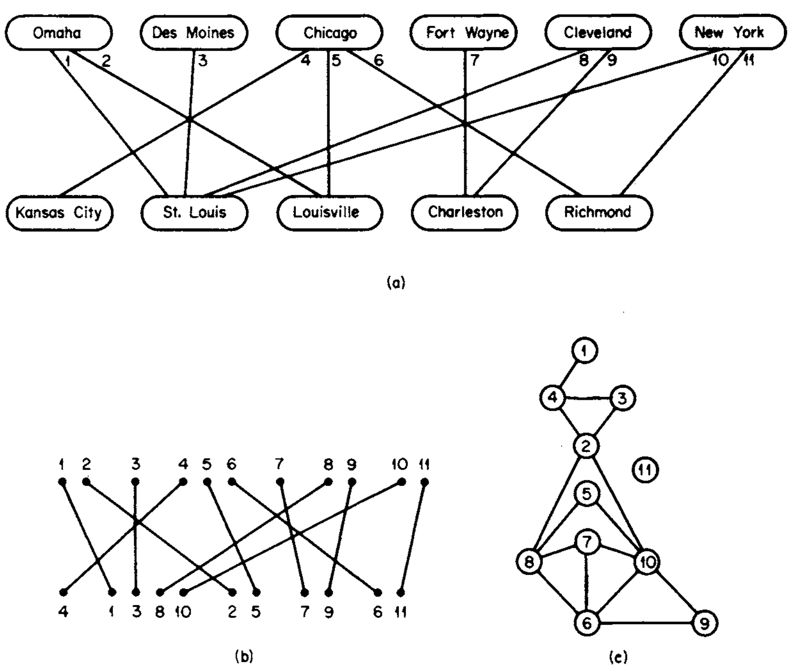


Figure 7.6. (a) A bipartite graph B representing flight paths between cities. (b) The matching diagram of a permutation π extracted from the bipartite graph B. (c) The graph $G[\pi]$. Color the vertices of $G[\pi]$ and solve the altitude assignment problem for B.

Application 7.2 (Shifted Intervals). Let $\mathscr{I} = \{I_i | i = 1, 2, \ldots, n\}$ be a collection of intervals on a line, where $I_i = (x_i, y_i)$ and $|I_i| = y_i - x_i$ denotes the length of I_i. Assume that the intervals, which may overlap, have been ordered such that $x_1 \leq x_2 \leq \cdots \leq x_n$. Let w_i represent the cost of shifting the interval I_i (assumed to be independent of the distance shifted). Find the cheapest shifting of intervals so that (1) the order is preserved and (2) no overlap remains. (In Even, Pnueli, and Lempel [1972], the intervals correspond to the memory requirements of n programs at a certain time in a multiprogramming computer.)

A solution to this problem is as follows. Consider the oriented graph $(\mathscr{I}, F)$ where

$$(I_i, I_j) \in F \Leftrightarrow \sum_{i \leq k \leq j} |I_k| \leq y_j - x_i \qquad (i < j).$$

Two intervals are thus related in F if and only if the intervals between them can be shifted in such a way that none of these $j - i + 1$ intervals (including the fixed I_i and I_j) will intersect. It is routine to show that F is a transitively oriented graph (see Exercise 5). The solution to our problem will be to find a chain of F having maximum weight (to remain, all others are shifted); in other words, find a maximum weighted clique of the graph $E = F + F^{-1}$ which is not only a comparability graph but is even a permutation graph.

5. Sorting a Permutation Using Queues in Parallel

A *queue* is a linear storage device in which items are loaded at one end and unloaded at the other end in a first-in-first-out fashion (FIFO). Let us consider the problem of sorting a permutation π of the numbers $1, 2, \ldots, n$

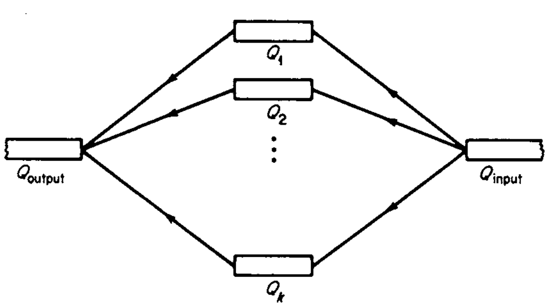


Figure 7.7. A network of k queues in parallel.

using a *network of k queues arranged in parallel* (see Figure 7.7). The permutation sits in the input queue initially. Each number, in turn, passes along to one of the k internal queues where it is stored temporarily until it is moved onto the output queue. We assume that each queue has unbounded capacity and that backing up along an edge, counter to its direction, is forbidden. One can easily imagine a station master directing railroad cars through such a switchyard in order to rearrange the cars of a freight train. Typically, the number of sidings (queues) will be limited, so we are led to the following problem. Given a network of k queues in parallel, characterize the permutations which can be sorted on it. Or similarly, given a permutation π, how many queues will we need? In addition, find an optimal sorting method.

Example. Suppose $\pi = [4, 3, 6, 1, 5, 2]$. The 4 is placed in Q_1. The 3 cannot go in Q_1 because it will be forever stuck behind the 4, so put it in Q_2. Next comes the 6, which can go either behind 4 on Q_1 or behind 3 on Q_2. Put 6 behind 4 on Q_1. How about 1? It must go on Q_3. The 5 cannot go on Q_1 because 6 is already there; put 5 on Q_2 behind 3. Finally 2 cannot go on Q_1 or Q_2, but it can go on Q_3. Now that everything is stored (Figure 7.8), we unload the numbers 1–6 from their respective storage places.

We call your attention to a few obvious facts. The contents of each Q_i must be in increasing order, for otherwise it would be impossible to successfully unload all the numbers in proper order. Furthermore, it is easy to show that it makes no difference whatsoever whether we (a) require loading *all* input numbers onto the queues before unloading any of them or (b) allow unloading anytime it is possible.

What is it that forces two numbers to go into different queues? Answer: The numbers occur in reversed order in π. Thus, if i and j are adjacent in $G[\pi]$, then they *must* go through different queues.

Proposition 7.3. Let $\pi = [\pi_1, \pi_2, \ldots, \pi_n]$ be a permutation of the integers $\{1, 2, \ldots, n\}$. There is a one-to-one correspondence between the

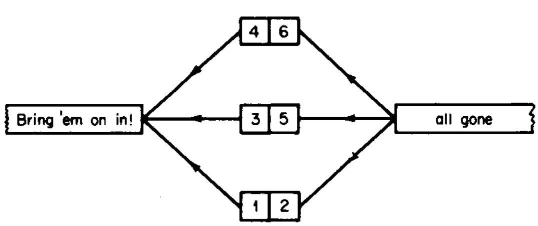


Figure 7.8. A network which is sorting $\pi = [4, 3, 6, 1, 5, 2]$.

proper k-colorings of $G[\pi]$ and the successful sorting strategies for π in a network of k parallel queues.

Proof. Assign painters to each Q_i, each with a different color paint. Now sort π in the k-network and have every number painted as it enters its corresponding queue. Since connected vertices i and j of $G[\pi]$ pass through different queues, they receive different colors.

Conversely, given a proper coloring of $G[\pi]$ using colors $1, 2, \ldots, k$, assign a traffic director to the input queue. If the color of x is c, then the traffic director sends x to Q_c. Suppose this strategy is unsuccessful. There must be a bottleneck in some queue, say Q_m; i.e., Q_m has a pair of numbers x and y stored in reversed order. However, x and y enter Q_m in the same order that they appear in π, namely, reversed. Thus, x and y are adjacent in $G[\pi]$, and yet they are both colored the same, a contradiction. Clearly, this correspondence is a bijection. ∎

Corollary 7.4. Let π be a permutation. The following numbers are equal:

(i) the chromatic number of $G[\pi]$,
(ii) the minimum number of queues required to sort π,
(iii) the length of a longest decreasing subsequence of π.

Proof. The equivalence of (i) and (ii) follows immediately from Proposition 7.3, and its proof suggests a method for transforming a solution of one problem into a solution of the other. Equality between (i) and (iii) holds since a longest subsequence of π corresponds to a maximum clique of $G[\pi]$, which will be of size $\chi(G[\pi])$ since permutation graphs are perfect. ∎

The *canonical sorting strategy* for π places each number in the first available queue. (Our example was done that way.) From this strategy, we obtain the *canonical coloring* of $G[\pi]$. The following algorithm simulates the process. It yields a minimum coloring.

Algorithm 7.1. Canonical coloring of a permutation.
Input: A permutation $\pi = [\pi_1, \pi_2, \ldots, \pi_n]$ of the numbers $\{1, 2, \ldots, n\}$.
Output: A coloring of the vertices of $G[\pi]$ and the chromatic number χ of $G[\pi]$.
Method: During the jth iteration, π_j is transferred onto the queue Q_i having smallest index i satisfying $\pi_j \geq$ last entry of Q_i (i.e., the first allowable Q_i). We do not *actually* save the entire contents of Q_i. Instead, an array LAST(i) holds the last number in Q_i. The counter k keeps track of the actual number of queues (colors) used. The entire algorithm is as follows:

```
   begin
1.   k ← 0;
2.   for j ← 1 to n do
       begin
3.       i ← FIRST allowable queue;
4.       COLOR(π_j) ← i;
5.       LAST(i) ← π_j;
6.       k ← max{k, i};
       end
7.   χ ← k;
   end
```

In order to execute statement 3 efficiently, a type of binary insertion can be used. One such subroutine is given in Figure 7.9. The next result shows the correctness of Algorithm 7.1.

Theorem 7.5. Let π be a permutation of the numbers $\{1, 2, \ldots, n\}$. The canonical coloring of $G[\pi]$, as produced by Algorithm 7.1, is a minimum coloring.

Proof. Clearly, Algorithm 7.1 produces a proper χ-coloring of $G[\pi]$. We must show that $\chi = \chi(G[\pi])$. It is sufficient to show that π has a decreasing subsequence of length χ. Consider the predecessor function p defined as follows: If $\mathrm{COLOR}(\pi_j) = i \geq 2$, then $\pi_{p(j)}$ equals the value of $\mathrm{LAST}(i-1)$ during the jth iteration. Clearly, $\pi_{p(j)} > \pi_j$ and $p(j) < j$ since it was $\pi_{p(j)}$ sitting on Q_{i-1} which forced π_j to go down to Q_i. Then

$$\pi_{j_1}, \pi_{j_2}, \ldots, \pi_{j_\chi}$$

where

$$\mathrm{COLOR}(\pi_{j_\chi}) = \chi$$

and

$$\pi_{j_{i-1}} = \pi_{p(j_i)} \qquad \text{for} \quad i = \chi, \chi - 1, \ldots, 2$$

is the desired decreasing subsequence. ∎

```
procedure FIRST allowable queue:
  begin
    i ← 1; t ← k + 1;
    until i = t do
      begin
          r ← ⌊(i + t)/2⌋;
        if π_j ≥ LAST(r)
          then t ← r;
          else i ← r + 1;
      end
    return i;
  end
```

Figure 7.9.

Remark. To find a minimum clique cover of $G[\pi]$, apply Algorithm 7.1 to the reversal π^ρ of π.

Algorithm 7.1 can be used to color any permutation graph G in time proportional to $n \log n$ provided we have the permutation π and the isomorphism $G \to G[\pi]$. Notice that this complexity is independent of the number of edges of G. If we do not have π, then we would revert to the coloring algorithm for comparability graphs.

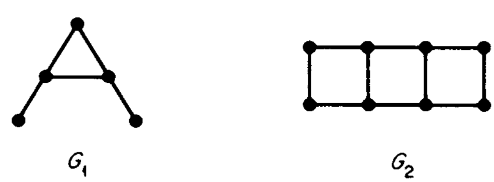
Remark. If we apply the algorithm in Exercise 8 of Chapter 2 to the orientation F of $G[\pi]$ where each edge is oriented toward its larger endpoint, then the coloring we obtain will be exactly the same as the canonical coloring, namely,

$$\mathrm{COLOR}(\pi_j) = i \Leftrightarrow \mathrm{HEIGHT}(\pi_j) = i - 1.$$

EXERCISES

1. For what permutation π of the numbers $1, 2, 3, \ldots, n$ is $G[\pi]$ the following:
 (a) the complete graph on n vertices;
 (b) the graph of n isolated vertices;
 (c) two disjoint complete graphs on r and $n - r$ vertices, respectively?

2. Find a permutation π whose graph $G[\pi]$ is isomorphic to G_1. Do the same for G_2 (see Figure 7.10).



G_1 G_2

Figure 7.10.

3. Let π^{-1} be the inverse of the permutation π. Prove that $G[\pi] \cong G[\pi^{-1}]$.

4. Let $\pi = [\pi_1, \pi_2, \ldots, \pi_n]$ be a permutation of $\{1, 2, \ldots, n\}$. Let p_i denote the number of integers less than and to the right of i in π, and let q_i denote the number of integers greater than and to the left of i in π. Prove that the following equality holds:

$$\pi_i^{-1} + p_i = i + q_i.$$

5. Let F be defined as in Application 7.2 (the shifted interval problem), and let $E = F \cup F^{-1}$.
 (i) Show that F is a transitive orientation of E.
 (ii) Prove that E is a permutation graph.

6. Give an application similar to Application 7.2 which uses the fact proved in Exercise 5(ii), namely, that E is not only a comparability graph, but is even a permutation graph.

7. A permutation graph G is *uniquely representable* if there is only one permutation π such that $G[\pi] \cong G$. Characterize the uniquely representable permutation graphs.

8. Let G be a permutation graph with n vertices. Given transitive orientations F_1 and F_2 of G and $\bar{G}$, respectively, write an algorithm which calculates a permutation π such that $G[\pi] \cong G$. Show that your algorithm can run in $O(n^2)$ time.

9. Using the canonical sorting strategy, give a minimum coloring of the graph $G[\pi]$ for $\pi = [9, 8, 2, 5, 6, 1, 7, 4, 3]$.

10. In sorting, using a network of parallel queues, prove that it makes no difference whether we (a) require loading *all* input numbers onto the queues before unloading any of them or (b) allow unloading anytime it is possible.

11. Prove the following: At any point during the execution of Algorithm 7.1,

$$\mathrm{LAST}(1) > \mathrm{LAST}(2) > \cdots > \mathrm{LAST}(k).$$

Why is this fact needed to justify the correctness of the subroutine in Figure 7.9? Analyze the time complexity of Algorithm 7.1.

12. Let G be a permutation graph with n vertices. Show that either G or $\bar{G}$ contains a clique of size $\lceil n^{1/2} \rceil$ (Erdös and Szekeres [1935]).

Bibliography

Alter, Ronald, Curtz, Thaddeus B., and Wang, Chung C.

[1974] Permutations with fixed index and number of inversions, *Proc. Fifth Southeastern Conf. on Combinatorics, Graph Theory and Computing*, Congressus Numerantium No. X, Utilitas Math, Winnipeg, 3–38. MR51 #2932.

Baker, Kirby A., Fishburn, Peter C., and Roberts, Fred S.

[1970] A new characterization of partial orders of dimension two, *Ann. N.Y. Acad. Sci.* **175**, 23–24. MR42 #140.

[1972] Partial orders of dimension 2, *Networks* **2**, 11–28. MR46 #104.

de Bruijn, N. G.

[1974] Sorting by means of swappings, *Discrete Math.* **9**, 333–339. MR49 #8869.

Erdös, P., and Szekeres, G.

[1935] A combinatorial problem in geometry, *Compositio Math.* **2**, 463–470.

Even, Shimon

[1973] "Algorithmic Combinatorics," Macmillan, New York. MR49 #48.

Even, Shimon, and Itai, Alon

[1971] Queues, stacks, and graphs, *in* "Theory of Machines and Computations," pp. 71–86. Academic Press, New York.

Even, Shimon, Pnueli, Amir, and Lempel, Abraham
[1972] Permutation graphs and transitive graphs, *J. Assoc. Comput. Mach.* **19**, 400–410. MR47 #1675.
Gill, M. K., and Acharya, B. D.
[1977] A new characterization of permutation graphs (unpublished).
Pnueli, Amir, Lempel, Abraham, and Even, Shimon
[1971] Transitive orientation of graphs and identification of permutation graphs, *Canad. J. Math.* **23**, 160–175. MR45 #1800.