Appendix

A. A Small Collection of NP-Complete Problems

GRAPH COLORING (decision version)
Instance: An undirected graph G and an integer $k > 0$.
Question: Does there exist a proper k-coloring of G? Equivalently, is $\chi(G) \leq k$?

GRAPH COLORING (optimization version)
Instance: An undirected graph G.
Question: What is $\chi(G)$?

CLIQUE (decision version)
Instance: An undirected graph G and an integer $k > 0$.
Question: Does there exist a complete subset of vertices of G of size k? Equivalently, is $\omega(G) \geq k$?

CLIQUE (optimization version)
Instance: An undirected graph G.
Question: What is $\omega(G)$?

STABLE SET (decision version)
Instance: An undirected graph G and an integer $k > 0$?
Question: Does G have a stable set of size k? Equivalently, is $\alpha(G) \geq k$?

STABLE SET (optimization version)
Instance: An undirected graph G.
Question: What is $\alpha(G)$?

CLIQUE COVER (decision version)
Instance: An undirected graph G and an integer $k > 0$.
Question: Can the vertices of G be covered by k cliques of G? Equivalently, is $k(G) \leq k$?

CLIQUE COVER (optimization version)
Instance: An undirected graph G.
Question: What is $k(G)$?

HAMILTONIAN PATH
Instance: An undirected graph G with vertices $v_1, v_2, \ldots, v_n$.
Question: Can the vertices be ordered $[v_{\pi_1}, v_{\pi_2}, \ldots, v_{\pi_n}]$ so that v_{π_i} and $v_{\pi_{i+1}}$ are adjacent in G for $i = 1, 2, \ldots, n - 1$?

HAMILTONIAN CIRCUIT
Instance: An undirected graph G with vertices $v_1, v_2, \ldots, v_n$.
Question: Can the vertices be ordered $[v_{\pi_1}, v_{\pi_2}, \ldots, v_{\pi_n}]$ so that v_{π_i} and $v_{\pi_{i+1}}$ are adjacent in G for $i = 1, 2, \ldots, n - 1$ and v_{π_n} and v_{π_1} are also adjacent in G?

STABLE SET ON TRIANGLE-FREE GRAPHS
Instance: An undirected graph G having no 3-cycle.
Question: What is $\alpha(G)$?

B. An Algorithm for Set Union, Intersection, Difference, and Symmetric Difference of Two Subsets

Input: Two subsets S and T of a universal set U whose members are numbered $u_1, u_2, \ldots, u_n$. All subsets are represented as lists of numbers (the indices of its members).
Output: The sets $S \cup T$, $S \cap T$, $S - T$, $T - S$, and $(S - T) \cup (T - S)$.
Method: An auxilliary Boolean n-vector $B = \langle b_1, b_2, \ldots, b_n \rangle$, initially containing only zeros, is used. As the list S is scanned, B is changed to the characteristic vector of S (line 3). In the loop 4–9, $S \cap T$ and $T - S$ are formed, $(S - T) \cup (T - S)$ is half formed, and B is changed to the characteristic vector of $S - T$. In the loop 10–15, $S \cup T$ and $(S - T) \cup (T - S)$ are completed and $S - T$ is formed. Also B is restored to the zero vector.

```
begin
1.  remark: $B = \langle 0, 0, \ldots, 0 \rangle$
2.  initialize: $S \cup T \leftarrow T$; $S \cap T \leftarrow S - T \leftarrow T - S \leftarrow (S - T) \cup (T - S) \leftarrow \emptyset$;
3.  for all $i \in S$ do $b_i \leftarrow 1$;;
4.  for all $j \in T$ do
5.    if $B_j = 1$
        then
6.        Add $j$ to $S \cap T$;
7.        $b_j \leftarrow 0$;
        else
8.        Add $j$ to $T - S$;
9.        Add $j$ to $(S - T) \cup (T - S)$;;
10. for all $i \in S$ do
11.   if $b_i = 1$
        then
12.       Add $i$ to $S \cup T$:
13.       Add $i$ to $S - T$;
14.       Add $i$ to $(S - T) \cup (T - S)$;
15.       $b_i \leftarrow 0$;;
16. remark: $B = \langle 0, 0, \ldots, 0 \rangle$
end
```

Complexity. Assuming no charge for initializing B (line 1), the complexity is dominated by the three loops. Thus, the algorithm runs in $O(|S| + |T|)$ steps.

C. Topological Sorting: An Example of Algorithm 2.4

Let us assume that the graph in Figure C1 is stored as sorted adjacency lists. Initially, the DFSNUMBER and the TSNUMBER of each vertex is

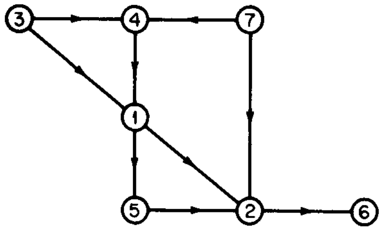


Figure C1

$$\text{TOPSORT}(1)\left\{\begin{array}{l}\text{DFSNUMBER}(1)\leftarrow 1\\ \text{TOPSORT}(2)\left\{\begin{array}{l}\text{DFSNUMBER}(2)\leftarrow 2\\ \text{TOPSORT}(6)\left\{\begin{array}{l}\text{DFSNUMBER}(6)\leftarrow 3\\ \text{TSNUMBER}(6)\leftarrow 7\end{array}\right.\\ \text{TSNUMBER}(2)\leftarrow 6\end{array}\right.\\ \text{TOPSORT}(5)\left\{\begin{array}{l}\text{DFSNUMBER}(5)\leftarrow 4\\ \text{TSNUMBER}(5)\leftarrow 5\end{array}\right.\\ \text{TSNUMBER}(1)\leftarrow 4\end{array}\right.$$

$$\text{TOPSORT}(3)\left\{\begin{array}{l}\text{DFSNUMBER}(3)\leftarrow 5\\ \text{TOPSORT}(4)\left\{\begin{array}{l}\text{DFSNUMBER}(4)\leftarrow 6\\ \text{TSNUMBER}(4)\leftarrow 3\end{array}\right.\\ \text{TSNUMBER}(3)\leftarrow 2\end{array}\right.$$

$$\text{TOPSORT}(7)\left\{\begin{array}{l}\text{DFSNUMBER}(7)\leftarrow 7\\ \text{TSNUMBER}(7)\leftarrow 1\end{array}\right.$$

Figure C2

set to 0, *j* is set to 7, and *i* is set to 0. The search begins with vertex 1. TOPSORT(1) will call TOPSORT(2), which will call TOPSORT(6); when control is eventually returned to TOPSORT(1) it will resume its scan of Adj(1) and will call TOPSORT(5). When TOPSORT(1) is finished, the main routine will call TOPSORT(3), etc. These recursive calls are illustrated in Figure C2. The final values of the depth-first search numbering and the topological sorting numbering are as follows:

Vertex	DFSNUMBER	TSNUMBER
1	1	4
2	2	6
3	5	2
4	6	3
5	4	5
6	3	7
7	7	1

D. An Illustration of the Decomposition Algorithm

The decomposition algorithm in Section 5.4 as applied to a noncomparability graph is illustrated in Figure D1.

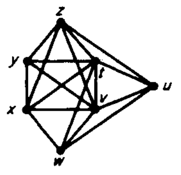
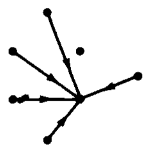
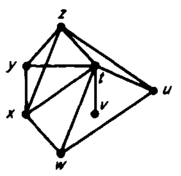
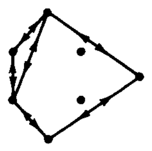
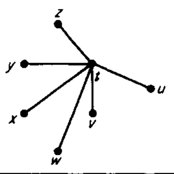
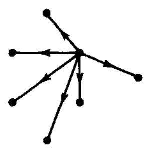
i	(V, E_i)	$a_i b_i$	(V, B_i)
1		yv	
2		zu	
3		tv	

Figure D1

E. The Properties P.E.B., C.B., (P.E.B.)', (C.B.)' Illustrated

Figure E1 gives examples of graphs satisfying or not satisfying the following properties:

P.E.B.:	the graph is a perfect elimination bipartite graph;
C.B.:	the graph is chordal bipartite;
(P.E.B.)':	the bipartite complement of the graph is perfect elimination bipartite;
(C.B.)':	the bipartite complement of the graph is chordal bipartite.

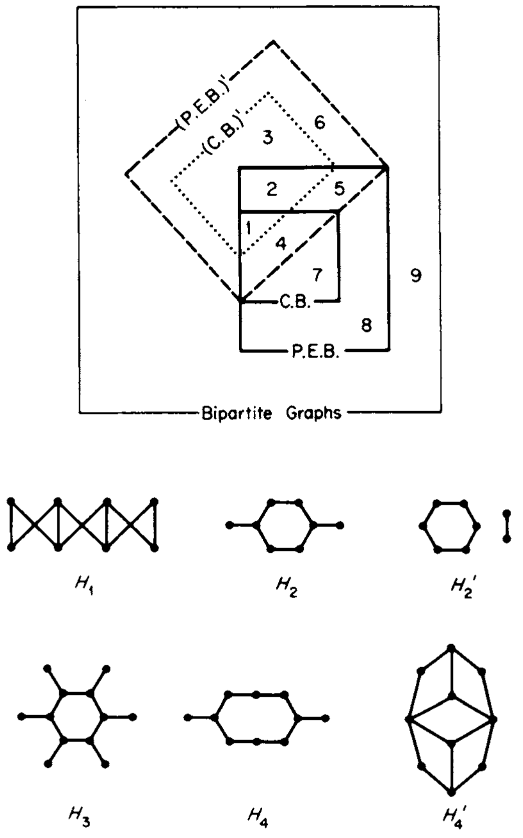


Figure E1

The regions in Figure E1 are illustrated by the given examples as follows:

Region	Example
1	H_1
2	H'_2
3	$3K_2$
4	H_2
5	H_3
6	H_4
7	C_6
8	H'_4
9	$C_n\ (n = 8, 10, 12, \cdots)$

F. The Properties C, $\bar{C}$, T, $\bar{T}$ Illustrated

Examples of graphs which are or whose complements are comparability graphs and/or triangulated graphs:

Property				
C	$\bar{C}$	T	$\bar{T}$	Examples*
+	+	+	+	Any threshold graph
+	+	+	−	$\bar{C}_4 = 2K_2$
+	+	−	+	C_4
+	+	−	−	G_6
+	−	+	+	G_5
+	−	+	−	G_4
+	−	−	+	$\bar{G}_2$ or $\bar{G}_3$
+	−	−	−	C_6, C_8, etc.
−	+	+	+	$\bar{G}_5$
−	+	+	−	G_2 or G_3
−	+	−	+	$\bar{G}_4$
−	+	−	−	$\bar{C}_6$, $\bar{C}_8$, etc.
−	−	+	+	G_7 or $\bar{G}_7$
−	−	+	−	$\bar{G}_1$
−	−	−	+	G_1
−	−	−	−	C_5, C_7, etc.

* See Figure F1 for the G_i.

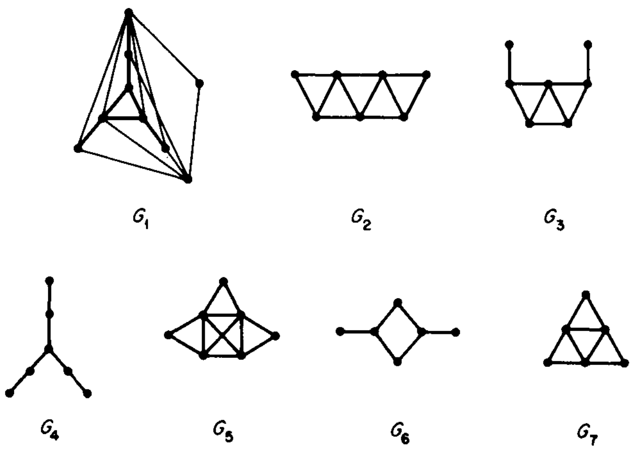


Figure F1