アルゴリズム 5−16
（1989. 1. 26）

有向グラフのk−連結成分を求めるアルゴリズムについて

牧野　正士
日本IBM（株）　東京基礎研究所

有向グラフGが、k+1個以上の頂点を持ち、かつ、任意の2頂点間に、両端点以外を共有しない有向道がk本以上あるとき、Gはk−連結であるという。一般に、Gがk−連結でないときでも、Gのある部分グラフがk−連結であることがあるが、Gのk−連結な部分グラフのうち極大なものをGのk−成分という。本稿では、与えられた整数k，l　（$0<k\leq l<n$）に対して、$k\leq \nu \leq l$を満たすGの全ての ν−成分を$O((n-k)\times T_l(G))$時間で求めるアルゴリズムについて詳述する。ここで、n，eは、それぞれGの頂点と辺の数、$T_l(G)$は、Gの最小分離点集合で、サイズがl以下のものを求める時間計算量で、$T_l(G)\geq n+e$　と仮定する。

On Digraph k-component Algorithms

Seishi Makino
IBM Research, Tokyo Research Laboratory
5-19, Sanbancho, Chiyoda-ku, Tokyo, 154, Japan

A digraph G is said to be k-connected if and only if it has at least $k+1$ vertices and there exist at least k vertex disjoint paths between every ordered pair of vertices. In general, G may have k-connected subgraphs, even if G itself is not k-connected. A maximal k-connected subgraph of G is called a k-component of G. We present an algorithm for finding all the ν -components of G which satisfies $1\leq k\leq \nu \leq l< n$, for any fixed k,l within $O((n-k)\times T_l(G))$ time, where n and e are the number of vertices and edges of G, respectively, and $T_l(G)$ is the time bound for finding a minimum vertex separator of G whose cardinality does not exceed l. We assume $T_l(G)\geq n+e$.

1. Introduction.

Consider a digraph (or graph) G without any self-loops or parallel-edges. We denote the vertex-set and edge-set of G by $V(G)$ and $E(G)$, respectively. We denote the cardinality of $V(G)$ and $E(G)$, namely, $|V(G)|$ or $|E(G)|$ by n and e, respectively. G is *k-connected*, iff $|V(G)| > k$ and there exist at least k vertex-disjoint paths from v to w, for every ordered pair of v and w ($v,w \in V(G)$). A *k-component* of G is a maximal k-connected subgraph of G (see Figure 1).

Hopcroft and Tarjan discovered algorithms for finding 1-components, 2-components and 3-components of a graph, as well as 1-components of a digraph, within $O(n+e)$ time [Ta72], [HoTa73]; however, no such linear algorithm has been found for $k \geq 4$, if G is a graph, nor for $k \geq 2$, if G is a digraph. Recently, Kanevsky and Ramachandran discovered an algorithm that finds all separating triplets of a 3-connected graph in $O(n^2)$ time [KaRa87], which suggests the existence of an efficient algorithm for finding 4-components of a graph.

Matula discovered a polynomial algorithm that finds all the k-components of a graph, using cluster analysis techniques [Ma77]. Note that the problems for graphs are easily reducible to those for symmetrical digraphs, although the converse is not true. Our purpose is to construct a polynomial time algorithm to find all the v-components ($k \leq v \leq l$) of a digraph for any fixed k,l. Our algorithm can be regarded as an extension and refinement (a practical, and probably more efficient version) of Matula's, and as essentially the same, but rather practical version of Makino's [M88].

Proceeding to our main results, we need to introduce some notations. For two finite sets, P,Q, we denote $P+Q=\{x|x \in P \text{ or } x \in Q\}$, and $P-Q=\{y|y \in P \text{ and } y \notin Q\}$. For $v,w \in V$, let $[v,w]$ denote a directed edge which leaves v and enters w. For $v \in V$, let $\Gamma^+(v) = \{w|\ [v,w] \in E(G)\}$, and $\Gamma^-(v) = \{w|\ [w,v] \in E(G)\}$. We denote $\deg(v) = \min\{\Gamma^+(v), \Gamma^-(v)\}$. G is *complete* if $\Gamma^+(v) = \Gamma^-(v) = n-1$, for $\forall v \in V(G)$. For $U(\subset V(G))$, $\ll U \gg_G$ denotes the *induced subgraph* of G by U, that is, a subgraph of G whose vertex-set is U, and whose edge-set is $\{[v,w] \in E(G)|v,w \in U\}$. For $S \subset V(G)$, we denote $\ll V(G)-S \gg_G$ as G/S. $S(\subset V(G))$ is a *separator* of G iff $V(G)/S$ is not 1-connected. For $l \geq 1$, a *l-separator*, denoted by $S^l(G)$, is a minimum separator, S, of G, such that $|S| \leq l$. Note that $S^l(G)$ is nothing but a minimum separator of G, if it exists, otherwise $S^l(G)$ is empty (see Figure 1). Then by the well-known Menger's theorem [Me27], G is k-connected iff $k \leq |S^l(G)|$ for some $l \geq k$. We assume all digraphs in our algorithm are represented by adjacency structures [AhHoUl74], [Ta72], and manipulated in an efficient way. Our results are summarized as follows.

Theorem A *Let G be a digraph with n vertices and e edges. All the possible v-components ($k \leq v \leq l$) of G are found within $O((n-k) \times T_l(G))$ time, where $T_l(G)$ is the time bound for computing $S^l(G)$. We assume $T_l(G) \geq n+e$.*

Many algorithm are available for computing $S^l(G)$ for digraphs [EvTa75],[Ga80]. In particular, Galil's approach [Ga80] yields an $O(l \times (n+e) \times \sqrt{n} \times Max\{l,\sqrt{n}\})$ time algorithm for computing $S^l(G)$. Thus we have:

Corollary 1 *All the possible v-components ($k \leq v \leq l$) of a digraph G are found within $O((n-k) \times l \times (n+e) \times \sqrt{n} \times Max\{l,\sqrt{n}\})$ time.*

Provided that G is a symmetrical digraph, or a graph, we can use more efficient algorithms for computing $S^l(G)$, [EsHa84], [GrHa86]. Combining Granot-Hassin's algorithm [GrHa86] with Even-Tarjan's maximum-flow algorithm [EvTa75], we can compute $S^l(G)$ in $O(n \times (n+e) \times min\{l,\sqrt{n}\})$ time. Thus we have:

Corollary 2 *All the possible v-components ($k \leq v \leq l$) of a symmetrical digraph (or graph) G are found within $O((n-k) \times n \times (n+e) \times min\{l,\sqrt{n}\})$ time.*

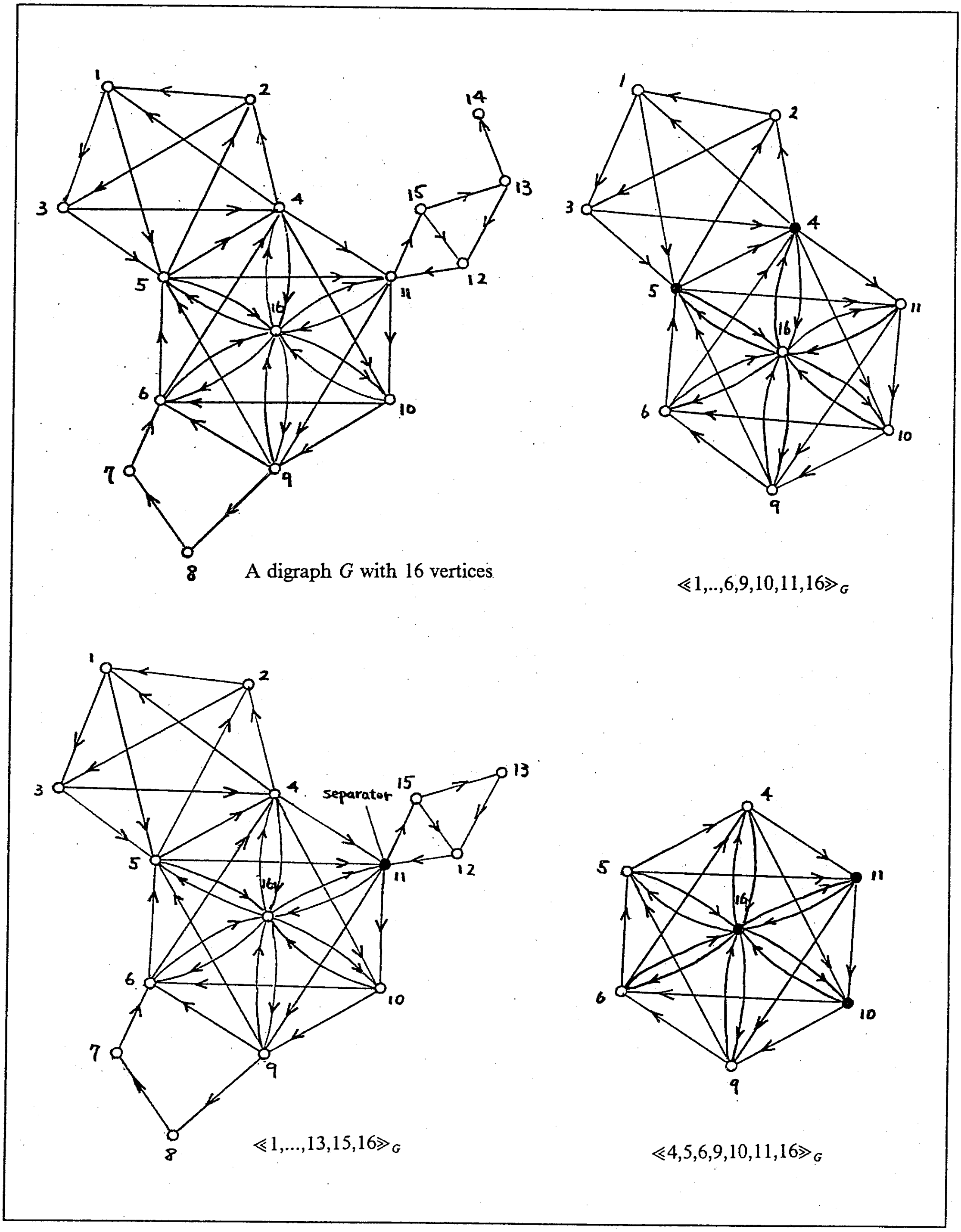

Figure 1. **A digraph G with 16 vertices.** G is not 1-connected. 1-component is $\ll 1,...,13,15,16 \gg_G$, 2-component is $\ll 1,..,6,9,10,11,16 \gg_G$, 3-component is $\ll 4,5,6,9,10,11,16 \gg_G$, $S^3(\ll 1,...,13,15,16 \gg_G)$ is $\{11\}$. $S^3(\ll 1,..,6,9,10,11,16 \gg_G)$ is $\{4,5\}$. $S^3(\ll 4,5,6,9,10,11,16 \gg_G)$ is $\{10,11,16\}$. $S^2(\ll 4,5,6,9,10,11,16 \gg_G)$ is empty.

Note that Corollary 1 and Corollary 2 show the same time bound, if $l \leq \sqrt{n}$. In particular, combined with Kanevsky-Ramachandran and Hopcroft-Tarjan algorithms [KaRa87], [HoTa73], Theorem 1 yields an $O(n^3)$ bound for finding 4-components of a graph.

Theorem B *For any k,n, s.t.* $1 \leq k < n$, *there exists a digraph with* $2 \times n - k + 1$ *vertices and* $2 \times n^2 - k \times (k-1)$ *edges which has every v-component* $(k \leq v \leq n)$ *with at least* $n+1$ *vertices and* $n \times (n+1)$ *edges.*

Theorem B shows that there are infinite class of digraphs with $O(n^2)$ vertices and $O(n^2)$ edges which require finding separator operations on $O(n-k)$ subgraphs, each of which has $O(n^2)$ vertices, for any fixed $k \leq n$. Therefore, time bound of theorem A is tight within a constant factor.

2. Algorithm.

```
procedure FMC(k,l,G);
begin if k ≤ l then
 begin if |V(G)| ≥ k + 1 then
  repeat
   for ∀v ∈ V(G) do
    if deg(v) < k then eliminate v, together with Γ+(v), and Γ-(v) from G;
    find 1-components of G;
    eliminate all edges linking between any two 1-components from G;
  until |V(G)| < k + 1, or min{deg(v)|v ∈ V(G)} ≥ k;
  if |V(G)| ≥ k + 1 then begin
    find 1-components of G;
   for each 1-component G_i of G do
    if |V(G_i)| ≥ k + 1 then
    if G_i is complete then output G_i as a (|V(G_i)| − 1)-component
    else if |V(G_i)| > k + 1 then begin
      find S^l(G_i);
      if S^l(G_i) is empty then output G_i as a (l + 1)-component
      else if|S^l(G_i)| ≥ k then begin
        k'←Max{k,|S^l(G_i)| + 1};
        output a copy of G_i as a |S^l(G_i)|-component;
        G_i←≪V(G_i) − S^l(G_i)≫_{G_i};
        find 1-components of G_i; any two 1-components from G_i;
        for each 1-component P_i^j of G_i do begin
           G_i^j←≪V(P_i^j) + S^l(G_i)≫_{G_i};
           FMC(k',l,G_i^j);
        end;
     end;
  end;
 end;
end;
end.
```

3. Proof of the theorems.

Eight lemmas are used in the proof of theorem A. The first five witness the correctness of $FMC(k,l,G)$, and the last three evaluates the time complexity.

Lemma 1 *l-components of a digraph are disjoint.*

Proof Let M, N be 1-components of a digraph G. Suppose M and N ($M \neq N$) have a vertex v in common. Then by definition of 1-components, there exists a path from u to v, and a path from v to w for any $u \in V(M)$ and $w \in V(N)$, which implies there exists a path from u to w in $\ll V(W) + V(N) \gg_G$. Similarly, there exists a path from w to u for any $w \in V(N)$ and $u \in V(M)$. Thus $\ll V(M) + V(N) \gg_G$ must be a 1-component of G, which contradicts the maximality of M and N.

Lemma 2 *If* $\deg(v) < k$, *then v is not contained in any k-component.*

Proof Immediate from the definition of the k-component.

Lemma 3 *Let G be l-connected digraph with* n (≥ 3) *vertices. If G is not complete, and* $|S^l(G)| > 0$, *there exist two vertices, v,w , such that, there exist no path from v to w in* $G/S^l(G)$.

Proof If G is not complete, there exist a vertex w, such that, $\deg(w) < n-1$, which implies $|S^l(G)| \leq n-2$. Therefore, $G/S^l(G)$ has at least two vertices. On the other hand, $G/S^l(G)$ is disconnected. Thus, there exist two vertices, v,w, such that there exist no path from v to w in $G/S^l(G)$. Q.E.D.

Lemma 4 *Let* $H \subset G$, *Then H is a k-component* ($k \leq l$) *of G if and only if* $|S^l(H)| \geq k$ *and* $|S^l(I)| < k$ *for every I which satisfies* $H \subset I \subset G$ *and* $H \neq I$

Proof Immediate from the definition of the k-component.

Lemma 5 *Let G be l-connected digraph, and let* $P^1, ..., P^N$ *be the l-components of* $G/S^l(G)$. *Then, every k-component* ($k > |S^l(G)|$) *of G is a subgraph of* $\ll V(P^i) + S^l(G) \gg_G$, *for some unique* i ($1 \leq i \leq N$). *Conversely, every k-component* ($k > |S^l(G)|$) *of* $\ll V(P^i) + S^l(G) \gg_G$ *is a k-component of G.*

Proof Suppose H be a k-component ($|S^l(G)| < k$) of G. Let $v,w \in V(H)$, and $v,w \notin S^l(G)$. Then there exist a path from v to w, and a path from w to v, each of which avoids $S^l(G)$, because there are at least k vertex disjoint paths both from v to w and from w to v, by Menger's theorem, which implies $v, w \in V(P^i)$ for some unique i ($1 \leq i \leq N$). Thus, $V(H) \subset V(P^i) + S^l(G)$, for some unique i. Therefore, $H \subset \ll V(P^i) + S^l(G) \gg_G$, for some unique i ($1 \leq i \leq N$). Since $\ll V(P^i) + S^l(G) \gg_G \subset G$, the maximality of H is obvious. Conversely, suppose I be a k-component of $\ll V(P^i) + S^l(G) \gg_G$ Then, since I is a k-connected subgraph in G, $I \subset I'$ holds for some k-component, I', of G. Then by the previous argument, I' must be a k -component of $\ll V(P_i) + S^l(G) \gg_G$. Therefore, by the maximality of I, $I = I'$. Q.E.D.

Note that Lemma 3 implies each $\ll V(P^i) + S^l(G) \gg_G$ in lemma 5 has fewer vertices than G, unless $S^l(G)$ is empty.

Lemma 6 *Overall time complexity to find* $S^l(G)$ *in line 13 of* $FMC(k,l,G)$, *including recursive calls is bounded by* $c \times (n-k) \times T_l(G)$, *where* c (≥ 1), *and* $T_l(G)$ *is the time bound to find* $S^l(G)$. *We assume* $T_l(G) \geq n+e$.

Proof Let $|V(G)| = n$, $|V(G_i)| = n_i$, $|V(G_i')| = n_i'$, and $|S^l(G)| = s_i$. We prove this by the induction on $m = n-k$.

1) $m = 1$: trivial.

2) Assume the lemma holds for $\forall 1 \leq m \leq M$. Consider the case, $m\,(=n-k)=M+1$. Since G_i are disjoint by Lemma 1, time complexity to find $S'(G_i)$ in line 16, excluding recursive calls, is bounded by:

$$\sum_i T(G_i) \leq T(G).$$

Next, we show that time complexity of the recursive calls, $FMC(k',l,G_i^j)$, in line 25 for each i, is bounded by:

$$c \times T(G_i) \times (n_i - k - 1).$$

At line 16, $|V(G_i)| \geq 3$, and G_i cannot be complete. Hence by lemma 3, $n_i^j \leq n_i - 1 \leq n-1$. Note that $k' \geq k$. Then:

$$n_i^j - k' \leq n - k - 1 \leq M.$$

Therefore, by the induction assumption, each $FMC(k',l,G_i^j)$ spends at most $c \times (n_i^j - k') \times T(G_i^j)$ time. Suppose G_i (at line 22) has N 1-components. Since $T(G_i^j) \leq T(G_i)$, time complexity for the executions of $FMC(k',l,G_i^j)$ for all j is bounded by:

$$\sum_{j=1}^{N} c \times (n_i^j - k') \times T(G_i^j) \leq c \times T(G_i) \times \sum_{j=1}^{N} (n_i^j - k'). \qquad \text{(A)}$$

Recall $k' \geq k$, and $n_i^j \leq n_i - 1$.
If $N=1$,

$$\text{(A)} \leq c \times T(G_i) \times (n_i^j - k') \leq c \times (n_i - k - 1) \times T(G_i).$$

Note that $|S'(G_i)| + \sum_j |V(P_i^j)| \leq n_i$. Then:

$$\sum_{j=1}^{N} n_i^j = \sum_{j=1}^{N} \{|V(P_i^j)| + |S'(G_i)|\} \leq n_i + (N-1) \times s_i.$$

Note that $k' \geq s_i + 1$, and $k' \geq k$.
If $N \geq 2$,

$$\begin{aligned}\text{(A)} &\leq c \times T(G_i) \times \{n_i + (N-1) \times s_i - (N-1+1) \times k'\} \\ &= c \times T(G_i) \times \{n_i - k' - (N-1) \times (k' - s_i)\} \\ &\leq c \times T(G_i) \times (n_i - k - 1).\end{aligned}$$

Thus the overall time complexity to find $S'(G_i)$, including recursive calls, is bounded by:

$$\sum_i \{T(G_i) + c \times (n_i - k - 1) \times T(G_i)\} \leq c \times (n-k) \times T(G).$$

Therefore, by the induction, the lemma holds for $\forall m \geq 1$. Q.E.D.

Lemma 7 *Time complexity of the iteration in lines 4-9, including recursive calls, is bounded by* $c \times (n-k) \times (n+e)$ *time, for some* $c\ (\geq 1)$.

Proof Let $m = n-k$. We prove by the induction on m.

1) $m=1$: Obvious.

2) Assume the lemma holds for $\forall 1 \leq m \leq M$. Let $m\,(=n-k)=M+1$.
In the same manner as Lemma 6, using the induction assumption, it is easily shown that the time complexity for each G_i (recursive part, concerning this iteration) is bounded by: $c \times (n_i - k - 1) \times (n_i + e_i)$.

Let the iteration (lines 4-9) be repeated r times, excluding recursive calls. Then, at least $r-1$ vertices are eliminated from G by the iteration. Therefore, after the iteration, it becomes: $\sum_i n_i \leq n-r+1$. Thus, the overall time complexity is bounded by:

$$r \times c \times (n+e) + \sum_i \{c \times (n_i - k - 1) \times (n_i + e_i)\} \leq r \times c \times (n+e) + c \times (n-r+1-k-1) \times (n+e)$$

$$= c \times (n-k) \times (n+e) .$$

Q.E.D.

Lemma 8 *FMC(k,l,G) is called at most $(n-k)$ times, including itself.*

Proof Proof is by the induction on $n-k$, same as Lemma 5. Q.E.D.

Theorem A *Let G be a digraph with n vertices and e edges. All the possible v-components $(k \leq v \leq l)$ of G are found within $O((n-k) \times T_l(G))$ time, where $T_l(G)$ is the time bound for computing $S^l(G)$. We assume $|T_l(G)| \geq n+e$.*

Proof The correctness easily follows from lemmas 1 - 5. In each execution of *FMC(k,l,G)*, processes except for what are evaluated in Lemma 6 and Lemma 7, spend at most $c' \times (n+e)$ time, excluding recursive calls, for some c' (>0). Since *FMC(k,l,G)* is called at most $(n-k)$ times by Lemma 8, these processes spend at most $c' \times (n-k) \times (n+e)$ time. Thus, the overall time complexity is bounded by:

$$c \times (n-k) \times T_l(G) + c \times (n-k) \times (n+e) + c' \times (n+e)$$
$$= O((n-k) \times T_l(G)) \text{ time.}$$ Q.E.D.

Theorem B *For any k,n, s.t. $1 \leq k < n$, there exists a digraph with $2 \times n - k + 1$ vertices and $2 \times n^2 - k \times (k-1)$ edges which has every v-component $(k \leq v \leq n)$ with at least $n+1$ vertices and $n \times (n+1)$ edges.*

Proof Construct G as follows.

(Stage 1) Let G_1 be a complete digraph with $n+1$ vertices, v_1 , ... ,v_{n+1}.

(Stage i ,$2 \leq i \leq n-k+1$) Let G_i be a digraph constructed by adding a new vertex v_{n+i} and new edges, $[v_{n+i}, v_j]$, and $[v_j, v_{n+i}]$ $(1 \leq j \leq n-i+1)$ to G_{i-1}.

(Stage $n-k+2$) Let $G = G_{n-k+1}$.

Note that G_1 has $n+1$ vertices and $n \times (n+1)$ edges, and in each stage i, a new vertex and $2 \times (n-i+1)$ edges are added to G_{i-1}. Therefore, G has $2 \times n - k + 1$ vertices and, the number of edges is:

$$n \times (n+1) + \sum_{i=2}^{n-k+1} 2 \times (n-i+1) = 2 \times n^2 - k \times (k+1).$$

Clearly, $|V(G_i)| \geq |V(G_1)| = n+1$, and $|E(G_i)| \geq |E(G_1)| = n \times (n+1)$. By a simple observation,

$$\{v \in V(G) | \deg(v) \geq j\} = \{v_j | 1 \leq j \leq 2 \times n - j + 1\} = G_{n-j+1} \quad (k \leq j \leq n).$$

For $k \leq j \leq n$, G_{n-j+1} is clearly j -connected, since G_{n-j+1} may not be disconnected by the removal of any p vertices $(p < j)$. Thus, G_{n-j+1} is the j-component of G $(k \leq j \leq n)$. Q.E.D.

4. Applications.

Our algorithm may provide a heuristic for certain kinds of induced subgraph problems. We define the induced subgraph problem as follows:

Given a graph (or digraph) G, positive integer I ($\leq |V(G)|$), and a property $\wp$. Is there a subset $W \subset V(G)$ with $|W| \geq I$ such that $\ll W \gg_G$ satisfies $\wp$?

In case $\wp$ means "clique", "independent set", "planar", "bipartite", "outerplanar", "edge graph", "chordal", "comparability graph", or "forest", the induced subgraph problem is known to be NP-complete [GaJo79]. However, our result shows that the induced subgraph problem is solvable in polynomial time, if $\wp$ means "k-connected". Note that if given graph (or digraph) G has a clique with k (≥ 2) vertices, then G has a $(k-1)$-component. Let k^* denote the maximum value of k over all the k-components of G, that is, G has a k^* -component, but no (k^*+1) -component. Clearly, our algorithm determines k^* in polynomial time. On the other hand, if $I > k^*+1$, then G cannot have any clique with no less than I vertices. Therefore the induced subgraph problem for "clique" with such I would be answered "No".

5. Remarks.

In our algorithm, degree checks and eliminations of unavailable edges in lines 4 - 9 are not necessary to achieve the time bounds of theorem A. However, in many cases, these simple checks may yield considerable reduction of the cost, while they may cause no effect in the worst case as described in theorem B.

6. Acknowledgement

I would like to thank Prof. Takao Asano of Sophia University for his valuable comments, and Dr. Kazuo Iwano and Dr. Takeshi Tokuyama of IBM Japan for helpful discussions. I also wish to thank Dr. Yoshihiro Akiyama of IBM Japan for his encouragement.

7. References.

[AhHoUl74] A.V. Aho, J.E. Hopcroft and J.D. Ullman, *The Design and analysis of computer algorithms, Addison-Wesley, Reading, MA, 1974.*

[EsHa84] A.H. Esfhanian and S. L. Hakimi, *On computing the connectivities of graphs., Networks 14, 1984, 355-366.*

[EvTa75] S. Even and R.E. Tarjan, *Networkflow and testing graph connectivity, SIAM. J. Comput.4, No.4, 1975, 507-518.*

[Ga80] Z. Galil, *Finding the vertex connectivity of graphs, SIAM. J. Comput. 9, No.1, 1980, 197-199.*

[GaJo79] M. Garey and D. Johnson, *Computers and Intractability: A Guide to Theory of NP-Completeness, Freeman, San Francisco, 1979.*

[GrHa86] F. Granot and R. Hassin, *Multiterminal maximum flows in node capacitated networks, Discrete Appl. Math. 13, Nos.2,3, 1986, 157-163.*

[HoTa73] J.E. Hopcroft and R.E. Tarjan, *Dividing a graph into triconnected components, SIAM. J. Comput. 2, No.3, 1973, 135-158.*

[KaRa87] A. Kanevsky and V. Ramachandran, *Improved algorithms for graph four-connectivity, FOCS, IEEE, 1987, 252-259.*

[M88] S. Makino, *An algorithm for finding all the k-components of a digraph, Intern. J. Comput. Math., 24, No.3+4, 1988, 213-221.*

[Ma77] D. Matula, *Graph theoretic techniques for cluster analysis algorithms, F. Van Ryzon, ed., Classification and Clustering, Academic Press, New York, 1977.*

[Me27] K. Menger, *Zur allgemeinen Kurventheorie, Fund. Math. 10, 1927, 96-115.*

[Ta72] R. Tarjan, *Depth-first search and linear graph algorithms, SIAM. J. Comupt., 1, No.2, 1972, 146-160.*