Discrete Mathematics	25/03/2019
Lecture 15: Paths, Components and Cycles	
Instructor: Goutam Paul	*Scribe: Ratnodeep Bain*

Contents

Introduction

Walk is just a sequence of edges that starts at a vertex and then travels from vertex to vertex. Many real life problems can be modeled with walks traveling along the edges of a graph. For example, the problem of efficiently planning routes for mail delivery (so that the delivery man has to travel minimum distance to deliver all the mails) can be solved using graph models that involve walks. In the next section we will deal with some basic terminologies related to walks.

1 Basic Terminologies

We begin with the definition of a walk. Let, $G = (V, E)$ be a graph.

Definition 1.1 (Walk)**.** *A **walk**, $w = (v_0, e_1, v_1, e_1, v_2, \ldots, e_k, v_k)$ is an alternate sequence of vertices and edges that begins and ends with vertices such that $e_i \in E$ is an edge between v_{i-1} and $v_i \in V$, for $i = 1, 2, 3, \ldots, k$.*

Remark. A walk of length k contains total n many ***edges*** in it. For example, the walk just defined is a walk of length k starting from v_0 to v_k. Note that a path of length zero consists of a single vertex.
When the graph is simple, we denote this path by its vertex sequence $v_0, v_1, \ldots, v_k$ (since we are considering only simple graphs, so vertex sequence uniquely determines edge sequence. Listing these vertices uniquely determines the path).

Definition 1.2 (Trail)**.** *A **trail** is a walk which does not repeat any edges.*

Definition 1.3 (Path)**.** *A **path** is a walk which does not repeat any vertices.*

Note.

1. *Every path is a trail but the converse is not true. A path has no repeated vertices so it can't repeat any edge since, repeating an edge will ensure repetition of at least two vertices.*

2. *The length of a path, cycle or walk is the number of edges in it.*

Example 1. *In the simple graph shown in* Figure 1*, a, d, c, f, e is a **walk** of length 4, because $\{a,\ d\}$, $\{d{,}c\}$, $\{c,\ f\ \}$, and $\{f,\ e\}$ are all edges. However, d, e, c, a is not a **walk**, because $\{e,\ c\}$ is not an edge.*

*Note that a, d, c, f, e is a **trail** of length 4 because, it has no repeated edge. The walk a, b, e, d, a, b, which is of length 5, is not a **trail** ,because it contains the edge $\{a, b\}$ twice.*

*a, d, c, f, e is also a **path** of length 4 because, it does not repeat any vertices.*

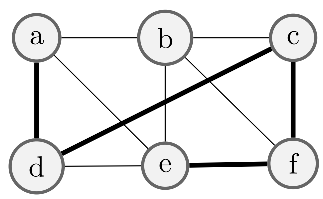

Figure 1

Now we will define <u>*closed walk*</u>, <u>*closed trail*</u> (also called *Circuit*) and <u>*closed path*</u> (also called *Cycle*).

2 Closed Walk, Circuit and Cycle

2.1 Basic Definitions

Definition 2.1 (Closed Walk)**.** *A walk* $w = (v_0, e_1, v_1, e_1, v_2, \ldots, e_k, v_k)$ *is called a* ***closed walk*** *if it starts and ends at the same vertex i.e* $v_0 = v_k$*.*

Definition 2.2 (Closed Trail)**.** *A* ***closed trail*** *or* ***circuit*** *is a trail with same start and end vertices. In other words it is a closed walk which does not repeat any edges.*

Definition 2.3 (Cycle)**.** *A* ***cycle*** *is a closed walk where no other vertices are repeated except the start and end vertices.*
Note that every cycle is a circuit. But every circuit is not a cycle.

Example 2. *In* Figure 2(a)*,* a, b, c, f, e, a *is a circuit of length* 5 *because, it begins and ends at* a *and it does not repeat any edges. It is a cycle also.*
But a, b, c, f, e, b, a *in* Figure 2(b) *is neither a circuit(because edge* $\{a, b\}$ *is repeated) nor a cycle(as* b *appeared twice in the walk). However, it ends where it started i.e* a*, so it a closed walk of length* 6*.*

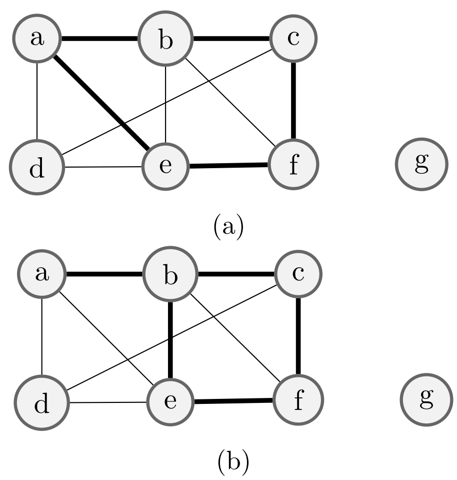

Figure 2

2.2 Results related to Minimum degree and Path-length, Cycle-length

Now We will discuss some results on minimum degree and path length.
Let, $G = (V, E)$ be a graph with set of vertices V and set of edges E. Suppose, v is an arbitrary vertex. Neighbourhood of v, $N(v)$=$\{u \in V | \exists$ an edge between $(u, v)\}$
Denote, $\delta(G) = min\{d(v) : v \in V(G)\}$ and $\Delta(G) = max\{d(v) : v \in V(G)\}$,where $d(v)$ is the degree of v.

Result 2.4. *Every graph G with minimum degree $\delta(G) \geq k$ contains a* <u>*path*</u> *of length* <u>*at least*</u> *k.*

Proof. For $k = 0$ the statement is trivial because, for any $v \in V$ the sequence v (of one term in V) forms a path of length 0.
So we assume that $k \geq 1$. Let n be the maximal possible length of a path in G, and let $v, v_1, v_2, \ldots, v_n$ be such a path(let's call that P) of length n. In order to prove the result, we have to show that $n \geq k$.

Claim 2.5. *Every $u \in N(v)$ lies on P.*

Proof. If not (thus the proof is conducted by contradiction), then $\exists\, u \in N(v)$ that does not lie on P. Then the sequence $u, v, v_1, v_2, \ldots, v_n$ forms a path of length $n+1$, a contradiction to the assumption that n is the maximal length of a path in the graph G. □

Now,

$$n = |P| \geq |N(v)| \geq \delta(G) \geq k$$

where the first and second inequality follows from the claim and the definition of $\delta(G)$, respectively. □

Result 2.6. *Every graph G with minimum degree $\delta(G) \geq k \geq 2$ contains a* <u>*cycle*</u> *of length* <u>*at least*</u> *$k + 1$.*

Proof. Let, P be a path of maximum length in G. Let, $P = (v_1, v_2, \ldots, v_n)$.
As in the previous proof, if v is a vertex adjacent with v_1, then $v \in \{v_2, v_3, \ldots, v_n\}$; else $v, v_1, v_2, \ldots, v_n$ is a path of greater length, which is a contradiction to the maximality of P. So, $N(v_1) \subseteq \{v_2, v_3, \ldots, v_n\}$. Let v_m be the last vertex in P to which v_1 is adjacent; see Figure 3. Then the sub-path $Q = (v_1, v_2, \ldots, v_m)$ contains at least $deg(v_1) + 1 \geq \delta(G) + 1$ vertices, and so $v_1, v_2, \ldots, v_m, v_1$ is a cycle of length $\geq \delta(G) + 1 \geq k + 1$.

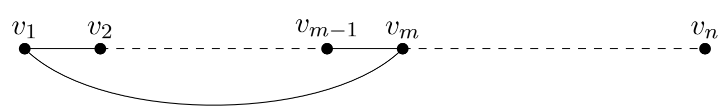

Figure 3

□

3 Connectivity

When does a computer network have the property that every pair of computers can share information, provided messages can be sent through one or more intermediate computers? When a graph is used to represent this computer network, where vertices represent the computers and edges represent the communication links, this question can be restated as : When is there always a path between two vertices in the graph?
If there is at least one path between two vertices, we call them connected. In this section firstly, we will discuss about connectedness in undirected graphs, then we will briefly introduce the notion of connectedness in directed graphs.

3.1 Path-connectedness Relation & Components

Definition 3.1 (u-v walk)**.** *Let, $G = (V, E)$ be an graph. u, v be any two vertices $\in V$. A **u-v walk** is defined as a sequence of vertices starting at u and ending at v, where consecutive vertices in the sequence are adjacent vertices in the graph.*

Definition 3.2 (u-v path)**.** *A **u-v path** is an u-v walk, where no vertex is repeated (each vertex is used at most once).*

Definition 3.3 (Connected vertices)**.** *In an undirected graph G, two vertices u and v are called **connected** if G contains a path from u to v. Existence of u-v path, implies connectivity between u and v.*

Definition 3.4 (Path-connectedness relation)**.** *Let, $G = (V, E)$ be an undirected graph. u, v be any two vertices $\in V$. Define a relation, R on V such that $(u, v) \in R$ if and only if u and v are connected.*

Theorem 3.5. *R is an equivalence relation on V.*

Proof. $\forall\, u \in V$, $(u, u) \in R$. Hence, R is reflexive by definition.
Assume that $(u, v) \in R$; then there is a path from u to v. Then $(v, u) \in R$ because there is a path from v to u, namely, the path from u to v traversed backward. So, R is symmetric.
Assume that $(u, v) \in R$ and $(v, w) \in R$; then there are paths from u to v and from v to w. Putting these two paths together gives a path from u to w. Hence, $(u, w) \in R$. Hence, R is transitive.
It follows that R is an equivalence relation. □

Note. *R induces a partition on the set of vertices since R is an equivalence relation on V. This parts are called connected components of the graph.*

Definition 3.6 (Connected Components)**.** *A **connected component** of a graph G is a connected sub-graph of G ,that is not a proper sub-graph of another connected sub-graph of G. That is, a connected component of a graph G is a maximal connected sub-graph of G.*

Each vertex belongs to exactly one connected component, as does each edge.

Result 3.7. *A graph with n vertices and m edges has at least $(n-m)$ connected components.*

Proof. Start with the empty graph (which has n components), and add edges one-by-one. Note that adding an edge can decrease the number of components by at most 1. Hence, the proof follows. □

Definition 3.8 (Connected Graph)**.** *An undirected graph $G = (V, E)$ is called* ***connected****, if number of connected components is* 1.
In other words, an undirected graph is called ***connected*** *if there is a path between every pair of distinct vertices of the graph.*

Example 3. *The graph G_1 in* Figure 4 *is connected, because for every pair of distinct vertices there is a path between them. G_1 has only one connected component i.e G_1 itself.*

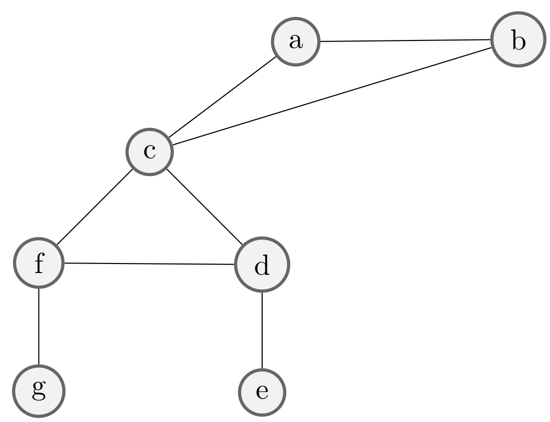

Figure 4: G_1 is a connected graph

Example 4. *Consider the graph, $G_2 = (V, E)$ where $V = \{a, b, c, d, 1, 2, 3, 4\}$ and $E = \{\{a, b\}, \{a, c\}, \{b, d\}, \{c, d\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}$.*
Note that G_2 has 3 maximal connected sub-graphs , which are - $\{a.b.c.d\}, \{1\}$ and $\{2, 3, 4\}$.
Clearly, G is not a connected graph.

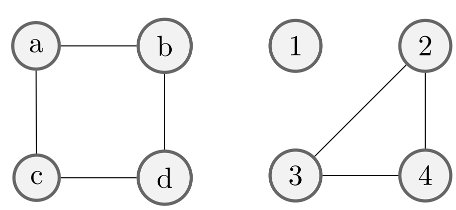

Figure 5: An example of connected components

Result 3.9. *There is a simple path between every pair of distinct vertices of a connected undirected graph.*

Proof. Let u and v be two distinct vertices of the connected undirected graph $G = (V, E)$. Because G is connected, there is at least one path between u and v. Let $x_0, x_1, \ldots, x_n$, where $x_0 = u$ and $x_n = v$, be the vertex sequence of a path of least length. This path of least length is simple. To see this, suppose it is not simple. Then $x_i = x_j$ for some i and j with $0 \leq i < j$. This means that there is a path from u to v of shorter length with vertex sequence $x_0, x_1, \ldots, x_{i-1}, x_j, \ldots, x_n$ obtained by deleting the edges corresponding to the vertex sequence $x_i, \ldots, x_{j-1}$. □

3.2 Connectedness in Directed Graphs

There are two notions of connectedness in directed graphs, depending on whether the directions of the edges are considered.

Definition 3.10 (Strongly connected)**.** *A directed graph, $G = (V, E)$ is* ***strongly connected*** *if there is a path from u to v <u>and</u> from v to u, whenever $u, v \in V$.*

Definition 3.11 (Weakly connected)**.** *A directed graph, $G = (V, E)$ is* ***weakly connected*** *if there is a path from u to v <u>or</u> from v to u, whenever $u, v \in V$.*

Clearly, any strongly connected directed graph is also weakly connected.

4 Bipartite and k-partite Graphs

Suppose that, there are m men and n women who desire to get married on an island. Each person has a list of members of the opposite gender acceptable as a spouse. Every woman can be married to at most one man, and every man to at most one woman. How could we marry everybody to someone they liked?

Let us try to model this problem in terms of a graph, $G = (V, E)$. We can consider each person willing to marry, to be a vertex in the graph so that, there is an edge between a man and a woman if they find each other acceptable as a spouse. Let, M and W be the set of vertices consisting of men and women respectively. Clearly, M and W are two disjoint proper subsets of V such that there union is the total set V. Another point to be noted is that there are no intra-edges between vertices of M or of W *i.e* M and W are independent sets of G.

Sometimes a graph has the property that its vertex set can be divided into two disjoint subsets such that each edge connects a vertex in one of these subsets to a vertex in the other subset. This leads us to next definition.

Definition 4.1 (Bipartite Graph)**.** *A* ***bipartite graph*** *is one whose set of vertices, V, can be divided into two independent sets, V_1 and V_2, and every edge of the graph connects one vertex in V_1 to one vertex in V_2 (Skiena 1990).*
When this condition holds, we call the pair (V_1, V_2) a ***bipartition*** *of the vertex set V of G.*

Now we will generalise this idea of bipartition to k-partition.

Definition 4.2 (k-partite)**.** *A graph $G = (V, E)$ is called* ***k-partite*** *if V can be partitioned into k independent sets and every edge of the graph connects one vertex in one of the independent sets to one vertex in another such sets.*

Example 5. *A cycle C_n, $n \geq 3$, consists of n vertices $v_1, v_2, \ldots, v_n$ and edges $\{v_1, v_2\}$, $\{v_2, v_3\}$,$\ldots$,$\{v_{n-1}, v_n\}$ and $\{v_n, v_1\}$. The cycle C_6 is displayed in* Figure 6.

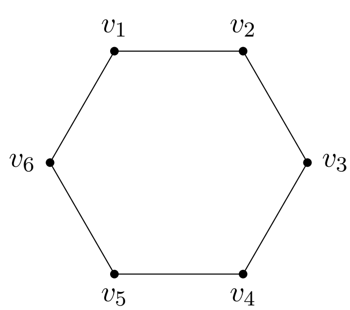

Figure 6: C_6

C_6 is bipartite, as shown in Figure 7*, because its vertex set can be partitioned into the two sets $V_1 = \{v_1, v_3, v_5\}$ and $V_2 = \{v_2, v_4, v_6\}$, and every edge of C_6 connects a vertex in V_1 and a vertex in V_2. Clearly, V_1 and V_2 are two independent set of V.*

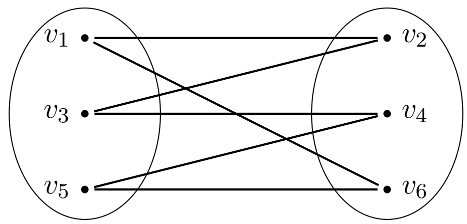

Figure 7: C_6 is bipartite

Example 6. *Graph $H = (V, E)$, as shown in* Figure 8*, is not bipartite because, its vertex set cannot be partitioned into two subsets so that edges do not connect two vertices from the same subset.*
Suppose that H is bipartite. Any one of the two partitions(wlog let's assume H_1) contain f, which is connected to every other vertices of H. Then H_1 can't contain any other vertices; otherwise it will not remain an independent set. So, $b \in H_2$. Similar arguments hold for b, as b is connected to all other vertices of H. But then $H_1 \cup H_2 = \{b, f\} \neq V$, which contradicts the initial assumption.

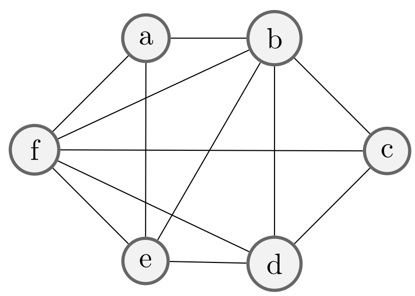

Figure 8: H is not bipartite

Test of bipartiteness of a graph is time consuming most of the times. We will prove a result that can be used to easily determine whether a graph is bipartite or not.

Result 4.3. *A graph is* ***bipartite*** *if and only if it doesn't contain any odd cycle.*

Proof.
Only if part: *Bipartite ⇒ No odd cycle*

If G is bipartite with bipartition X, Y of the vertices, then any cycle C has vertices that must alternately be in X and Y . Thus, since a cycle is closed, C must have an even number of vertices and hence is an even cycle.

If part: *No odd cycle ⇒ Bipartite*

It is enough to consider G as being connected, as otherwise we could consider each component separately.

Assume that $G = (V, E)$ has no odd cycle. If we can construct a bipartition of the vertex set V, the proof is done. Let $u \in V$ be an arbitrary vertex. Partition all other vertices based on the parity of distance (even or odd) from u. That is, let

$$V_1 = \{v \in V : \text{all shortest } u - v \text{ paths are odd}\}$$
$$V_2 = \{v \in V : \text{all shortest } u - v \text{ paths are even}\}$$

Now we will verify that this is indeed a partition.

Claim 4.4.

1. $V_1 \cap V_2 = \phi$
2. $V_1 \cup V_2 = V$.

Proof.

1. If not (thus the proof is conducted by contradiction), then $\exists\, v \in V_1 \cap V_2$.
   Now, $v \in V_1 \implies$ all shortest $u - v$ paths are odd.
   But, $v \in V_2 \implies$ all shortest $u - v$ paths are even.
   This leads to contradiction and hence completes the proof of the claim.

2. Since the graph is connected, for any arbitrary vertex $v \in V$, all the shortest $u - v$ paths are either of even (including zero, which happens to be the case when $v = u$) or odd length. So, any arbitrary vertex v is either an element of V_1 or V_2. Hence, $V_1 \cup V_2 = V$.

□

Claim 4.5.

1. *There is no edge between vertices of V_1.*
2. *There is no edge between vertices of V_2.*

Proof. We will prove the first statement. The second statement can be proved in a similar manner.

For the sake of contradiction let us assume that the claim is false.

Then there exists $s \neq t \in V_1$ such that $\{s, t\} \in E$. Since, s and t are distinct members of V_1, all shortest u-s and u-t paths are odd. Take any one such paths from each of them *i.e* P (of length $2p - 1$) from all shortest u-s paths and Q (of length $2q - 1$) from all shortest u-t paths. Suppose z is the last common vertex between P and Q. k be the length of the path from u to z. Then, lengths of the $z - s$ and $z - t$ paths are $2p - 1 - k$ and $2q - 1 - k$ respectively. Now, we can consider a cycle starting from z, that visits s,t(since, they are adjacent by assumption) and returns back to z.

Length of this cycle = $(2p - 1 - k) + 1 + (2q - 1 - k)$= $2(p + q - k) + 3$ is odd. But G can't have any odd cycles. Hence, the initial assumption was wrong and there is no edge between vertices of V_1. □

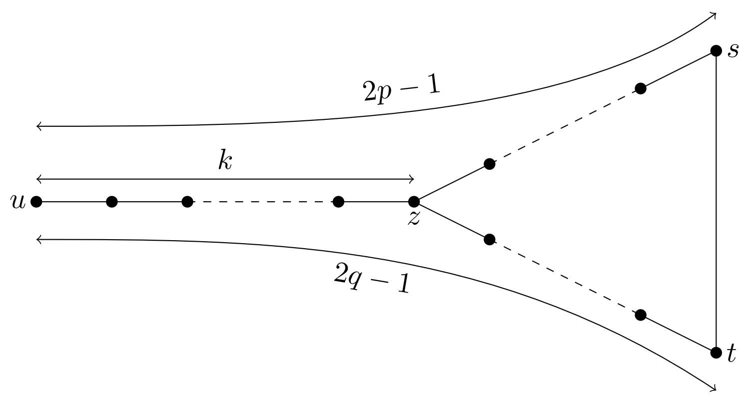

Figure 9

We have shown that V_1 and V_2 are partitions of V. So, G is bipartite.

Now consider G is not a connected graph. Suppose it has m connected components: $C_1, C_2, ..., C_m$. We can show that each of them is bipartite using similar arguments. Suppose that the bipartitions of $C_1, C_2, ..., C_m$ are (C_{1a}, C_{1b}) , $(C_{2a}, C_{2b}), .., (C_{ma}, C_{mb})$ respectively. Let, $V_1 = \bigcup_{i=1}^{m} C_{ia}$ and $V_2 = \bigcup_{i=1}^{m} C_{ib}$. Clearly, $V_1 \cap V_2 = \phi$, $V_1 \cup V_2 = V$ and there is no intra-edge between vertices of V_1 or V_2. So, G is bipartite.
This completes the "if part" of the proof. □

Example 7. *Now we can easily show that the graph H, shown in* Figure 8 *is not bipartite. It has many odd cycles. For example, f, a, b, f is an odd cycle of length 3. Hence we get, H is not a bipartite graph (using the previous result).*

5 Tournament, Kings and Associated Results

Definition 5.1 (Orientation of graph)**.** *Given an undirected graph G, each directed graph obtained from G by assigning some direction on each edge of G, is called **orientation** of G.*

Among directed graphs, the oriented graphs are the ones that have no 2-cycles (that is at most one of (x, y) and (y, x) may be arrows of the graph).

Definition 5.2 (Tournament)**.** *A **Tournament** is a directed graph (digraph) obtained by assigning a direction for each edge in an undirected complete graph. That is, it is an orientation of a complete graph, or equivalently a directed graph in which every pair of distinct vertices is connected by a directed edge with any one of the two possible orientations.*

Definition 5.3 (King)**.** *A vertex v in a tournament is called a **King** if every other vertices are reachable from v by a directed path of length of at most 2.*

If v is a King and u is any other vertex, then either v beats u or v beats someone who beats u.

Result 5.4. *Any maximum out-degree vertex must be a King.*

Proof. Suppose that v is a vertex of maximum out-degree in the tournament T. Then let A denote the out-neighborhood of v and let B denote the in-neighborhood of v. We claim that v is a King in T. To verify this claim, it is enough to show that there is a directed path of length 2 from v to each vertex in its in-Neighborhood.
So now suppose that u is any vertex in the in-neighborhood of v. If there is a vertex w in A such that $\{w, u\}$ is an edge of T, then v-w-u is a directed path of length 2 from v to u. On the other hand, if there is no such vertex in A, then it must be that u dominates every vertex of A(since, a tournament is obtained by assigning a direction for each edge in an undirected complete graph). Thus since u dominates everything that v does and u also dominates v, the out-degree of u is greater than that of v - but this contradicts the choice of v. Thus the claim is verified and the result holds. □

Remark. *All the maximum out-degree vertices are Kings, hence no non-King can have maximum out-degree.*

Its a natural question to occur in reader's mind that whether a given tournament has a King or not. We will answer this question by proving "**Every tournament has a King**".

Theorem 5.5. *Every tournament has a King.*

Proof. Let v be any vertex in the given tournament T. If v is a King ($i.e$ any vertex in T is reachable from v in at most 2 steps), then we are done.
Now, suppose v is not a King. We will give an algorithm for finding out a King in T starting from v. The proof will follow because from the last result we know that any maximum out-degree vertex is a King.
Let A denote the out-neighborhood of v and let B denote the in-neighborhood of v. Since,

v is not a King , then $\exists\ u \in B$ that is not reachable from v in at most 2 steps. So, for all $w \in A$, we can't go from w to u in one step. But the graph is complete, hence we can go from u to w in one step, $\forall\ w \in A$. u dominates every out-degree vertex of v as well as v itself. So, $d^+(u) \geq d^+(v) + 1$. If u is a King then we are done. Otherwise we continue until we get a King (exists as the tournament is finite).

□

Remark. *Note that in the previous algorithm we might end up getting a maximum out-degree vertex, which is King by* result 6.4. *Another important point to note that, "**A King may not be a maximum out-degree vertex**". We will discuss an example illustrating the former claim.*

Example 8. *Consider an orientation of the complete graph K_7 . We have shown only a few edges of the graph in* Figure 10 *to keep it simple. However, all the edges connected to a and g are drawn. Notice that if we start from vertex a or g, we can reach every other vertices by a directed path of length at most 2. So, a and g are Kings. But* $d^+(a) = 3$ *whereas* $d^+(g) = 5$. *Clearly, a is a King but not a maximum out-degree vertex.*

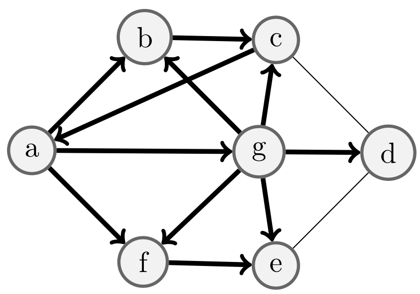

Figure 10: Example: "**A King may not be a maximum out-degree vertex**"