Locating sets and numbers for disconnected graphs

John Spaw

Millikin University

November 14, 2015

Overview

Background

Past Results

Computing locating numbers and sets using Sage

Disconnected Graphs

Graphs

Graphs

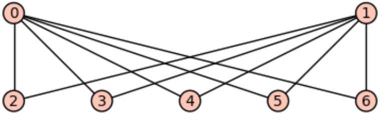
Definition

A **graph** $G(V, E)$ is a finite nonempty set $V(G)$ of *vertices* $V(G)$ together with a (possibly empty) set $E(G)$ of unordered pairs of distinct vertices called *edges*. [1]

Graphs

Definition

A **graph** $G(V, E)$ is a finite nonempty set $V(G)$ of *vertices* $V(G)$ together with a (possibly empty) set $E(G)$ of unordered pairs of distinct vertices called *edges*. [1]



Graphs

Graphs

Definition

A graph is **connected** if there exists a path between every two distinct vertices of $G(V, E)$. Otherwise, $G(V, E)$ is disconnected.

Graphs

Definition

A graph is **connected** if there exists a path between every two distinct vertices of $G(V, E)$. Otherwise, $G(V, E)$ is disconnected.

Definition

The **distance** $d(v_1, v_2)$ between two vertices v_1 and v_2 is the length of the shortest path from v_1 to v_2.

Complete graphs

Definition

A graph G is **complete** if there exists and edge between each two distinct vertices.

Complete graphs

Definition

A graph G is **complete** if there exists and edge between each two distinct vertices.

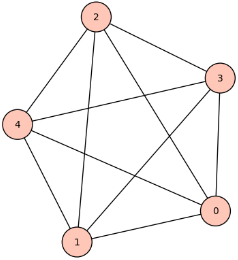


Figure: A complete graph on 5 vertices.

Disconnected graph example

Disconnected graph example

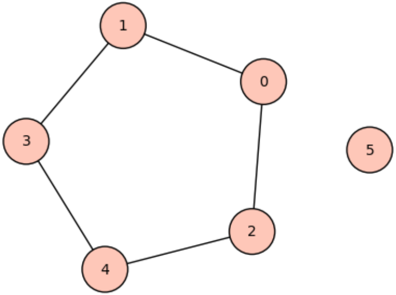


Figure: A disconnected graph.

Disconnected graph example

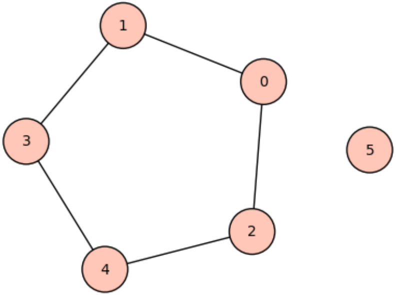


Figure: A disconnected graph.

$d(1, 2) = 2$

Disconnected graph example

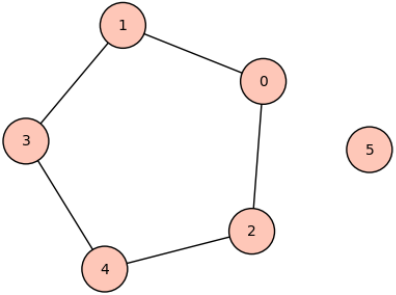


Figure: A disconnected graph.

$d(1,2) = 2$
$d(1,1) = 0$

Disconnected graph example

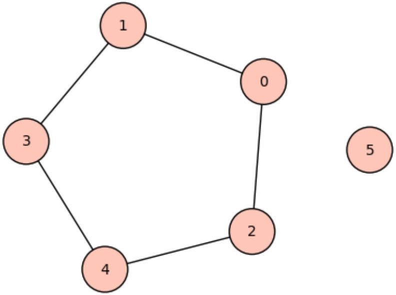


Figure: A disconnected graph.

$d(1,2) = 2$
$d(1,1) = 0$
$d(0,5) = \infty$

Development of locating sets and locating numbers

Development of locating sets and locating numbers

- Developed by Pirzada, Raja, and Redmond [1].

Development of locating sets and locating numbers

- Developed by Pirzada, Raja, and Redmond [1].
- A form of "triangulation" within a graph.

Development of locating sets and locating numbers

- Developed by Pirzada, Raja, and Redmond [1].
- A form of "triangulation" within a graph.
- Used for studying graphs of commutative rings.

Development of locating sets and locating numbers

- Developed by Pirzada, Raja, and Redmond [1].
- A form of “triangulation” within a graph.
- Used for studying graphs of commutative rings.
- Zero-divisor graphs (always connected).

Development of locating sets and locating numbers

- Developed by Pirzada, Raja, and Redmond [1].
- A form of "triangulation" within a graph.
- Used for studying graphs of commutative rings.
- Zero-divisor graphs (always connected).
- Locating sets only developed for *connected* graphs.

Locating sets and locating numbers

Locating sets and locating numbers

- Select an ordered subset W of the vertex set.

Locating sets and locating numbers

- Select an ordered subset W of the vertex set.
- Compare distances from vertices in $V(G)$ to each vertex in W.

Locating sets and locating numbers

- Select an ordered subset W of the vertex set.
- Compare distances from vertices in $V(G)$ to each vertex in W.
- Represent the distances as a vector called a **locating code**.

Locating sets and locating numbers

- Select an ordered subset W of the vertex set.
- Compare distances from vertices in $V(G)$ to each vertex in W.
- Represent the distances as a vector called a **locating code**.
- If W is a **locating set**, then each distinct vertex must have a distinct locating code.

Locating sets and locating numbers

Locating sets and locating numbers

Definition

Let G be a graph with n vertices. For an ordered set $W = \{w_1, w_2, \ldots, w_k\}$ of vertices of G and a vertex $v \in V(G)$, the **locating code** of v with respect to W is the k-vector

$$c_W(v) = (d(v, w_1), d(v, w_2), \cdots, d(v, w_k)).$$

Locating sets and locating numbers

Locating sets and locating numbers

Definition

The set W is a **locating set** for G if distinct vertices have distinct locating codes. A locating set with the minimum number of vertices is a **minimal locating set** for G.

Locating sets and locating numbers

Definition

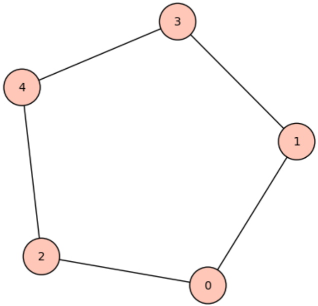
The set W is a **locating set** for G if distinct vertices have distinct locating codes. A locating set with the minimum number of vertices is a **minimal locating set** for G.

Definition

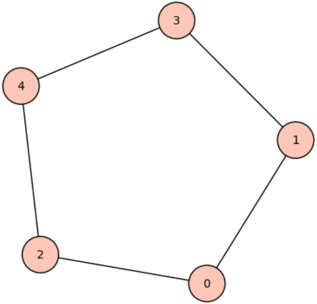
The **locating number** of G, denoted $loc(G)$ is the number of vertices in the minimal locating set.

Locating sets example

Locating sets example

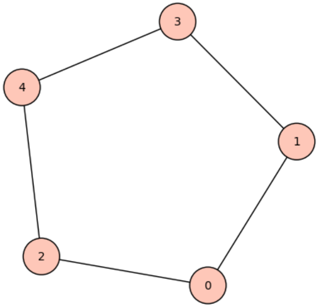


Locating sets example



Let $W = \{0, 1\}$

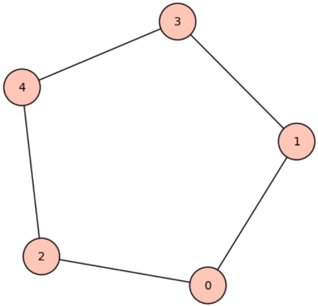
Locating sets example



Let $W = \{0, 1\}$

$c_W(0) = (0, 1)$

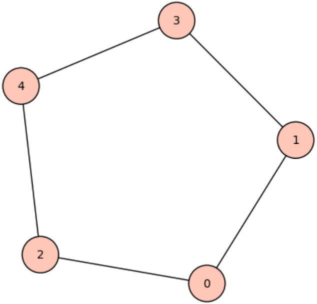
Locating sets example



Let $W = \{0, 1\}$

$c_W(0) = (0, 1) \qquad c_W(1) = (1, 0)$

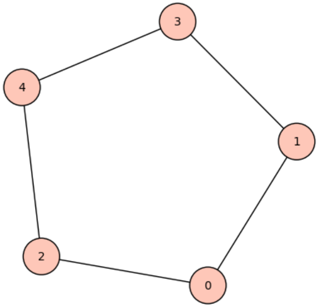
Locating sets example



Let $W = \{0, 1\}$

$c_W(0) = (0, 1)$ $\qquad$ $c_W(1) = (1, 0)$ $\qquad$ $c_W(2) = (2, 1)$,

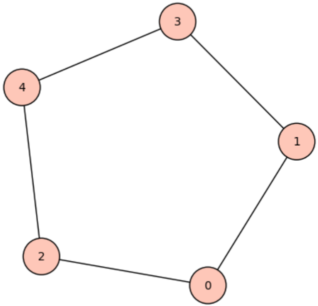
Locating sets example



Let $W = \{0, 1\}$

$c_W(0) = (0, 1)$ $\quad c_W(1) = (1, 0)$ $\quad c_W(2) = (2, 1)$,
$c_W(3) = (2, 2)$

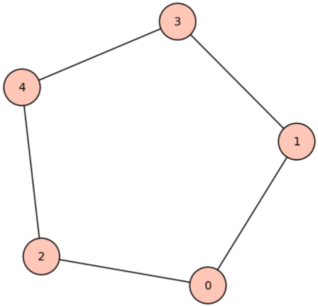
Locating sets example



Let $W = \{0, 1\}$

$c_W(0) = (0, 1)$ $\quad c_W(1) = (1, 0)$ $\quad c_W(2) = (2, 1)$,
$c_W(3) = (2, 2)$ $\quad c_W(5) = (1, 2)$

Locating sets example



Let $W = \{0, 1\}$

$c_W(0) = (0, 1)$ $c_W(1) = (1, 0)$ $c_W(2) = (2, 1)$,
$c_W(3) = (2, 2)$ $c_W(5) = (1, 2)$

W is a locating set for the graph because each locating code is distinct.

Locating numbers for specific graphs

Locating numbers for specific graphs

Pirzada et. al. found a classification of $loc(G)$ for *connected* graphs on n vertices in certain cases.

Locating numbers for specific graphs

Pirzada et. al. found a classification of $loc(G)$ for *connected* graphs on n vertices in certain cases.

- $loc(G) = n - 1$ if and only if G is complete.

Locating numbers for specific graphs

Pirzada et. al. found a classification of $loc(G)$ for *connected* graphs on n vertices in certain cases.

- $loc(G) = n - 1$ if and only if G is complete.
- $loc(G) = 1$ if and only if G is a path.

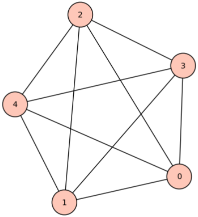
Locating numbers for specific graphs

Pirzada et. al. found a classification of $loc(G)$ for *connected* graphs on n vertices in certain cases.

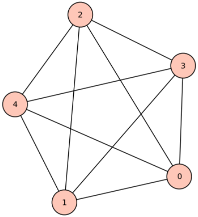
- $loc(G) = n - 1$ if and only if G is complete.
- $loc(G) = 1$ if and only if G is a path.
- $loc(G) = 2$ if and only if G is a cycle

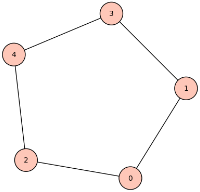
Examples:

Examples:

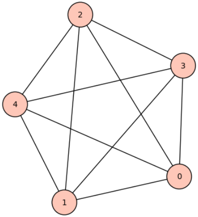


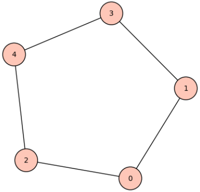
Examples:

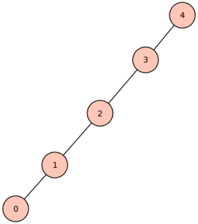




Examples:







Examples:

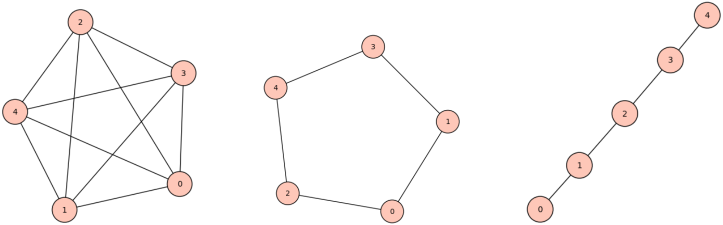


Figure: A complete graph A, a cycle B, and a path C, each on five vertices.

Examples:

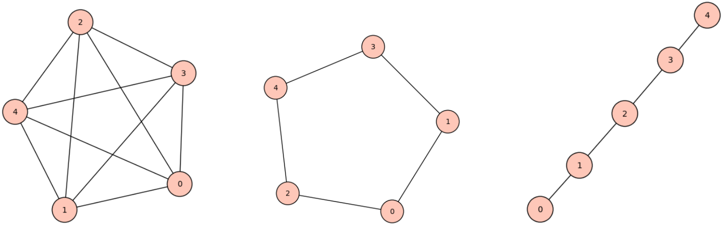


Figure: A complete graph A, a cycle B, and a path C, each on five vertices.

- $loc(A) = n - 1 = 4$

Examples:

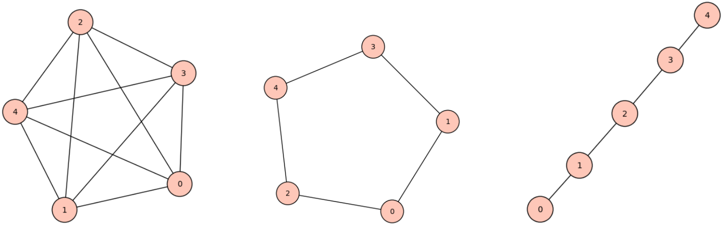


Figure: A complete graph A, a cycle B, and a path C, each on five vertices.

- $loc(A) = n - 1 = 4$
- $loc(B) = 2$

Examples:

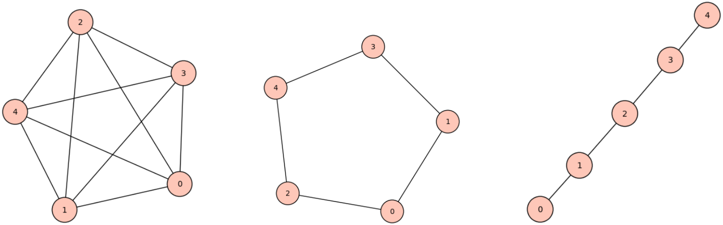


Figure: A complete graph A, a cycle B, and a path C, each on five vertices.

- $loc(A) = n - 1 = 4$
- $loc(B) = 2$
- $loc(C) = 1$

Disconnected graphs

Disconnected graphs

- Conditions for locating sets of disconnected graphs

Disconnected graphs

- Conditions for locating sets of disconnected graphs
- Locating numbers for disconnected graphs

Disconnected graphs

- Conditions for locating sets of disconnected graphs
- Locating numbers for disconnected graphs
- Generalized classification for past locating number results

Disconnected graphs

- Conditions for locating sets of disconnected graphs
- Locating numbers for disconnected graphs
- Generalized classification for past locating number results

Problem:

Disconnected graphs

- Conditions for locating sets of disconnected graphs
- Locating numbers for disconnected graphs
- Generalized classification for past locating number results

Problem:
As graphs become more complicated, computing locating sets becomes difficult.

Disconnected graphs

- Conditions for locating sets of disconnected graphs
- Locating numbers for disconnected graphs
- Generalized classification for past locating number results

Problem:
As graphs become more complicated, computing locating sets becomes difficult.

Let's have a computer do it for us!

Computing $loc(G)$ and minimal locating sets

Algorithm pseudocode:

Computing $loc(G)$ and minimal locating sets

Algorithm pseudocode:

```
for i in range(1,|V(G)|):
    combs = combinations of i vertices
    for k in combs
        compute locating codes with respect to k
        test each locating code for equality
            if two locating codes are equal:
                k is not a locating set
            else:
                k is a locating set
                print loc(G) = i
                print k
                break
```

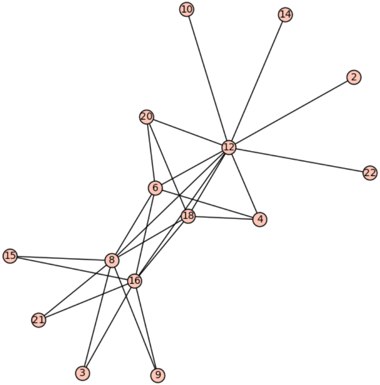
Computing locating numbers in Sage

Locating Number Algorithm

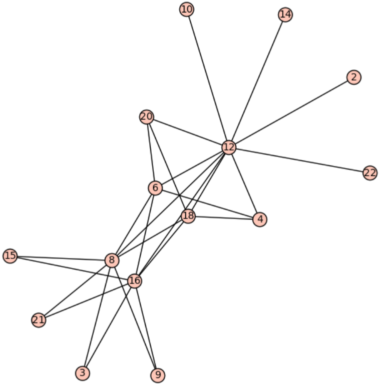
```
def locating_sets_num(A,G):          #Finds locating sets for a graph G with a vertex set A

    def make_unique(original_list):
        unique_list = []
        [unique_list.append(obj) for obj in original_list if obj not in unique_list]
        return unique_list

    for i in range(1,len(A)+1):
        combs=Combinations(A,i)

        locating_number=0                #Start at 0 by default
        locating_sets=[]                 #Contains all CONFIRMED locating sets
        for k in combs:                    #For each combination of vertices
            loc_set=[j for j in k]         #Array containing vertices from the locating set
            dist=[]                        #Array to hold all of the locating codes
            for vertex in G:
                code=[]
                for h in loc_set:
                    code.append(G.distance(vertex,h))
                dist.append(code)
            code_count=len(dist)
            duplicates=0
            for i in range(len(dist)):
                for j in range(len(dist)):
                    if j != i:
                        if dist[j] == dist [i]:
                            duplicates = duplicates + 1

            if code_count-duplicates == code_count:
                locating_sets.append(k)
            locating_sets = make_unique(locating_sets)
        if len(locating_sets) != 0:
            print('The following are locating sets: '+str(locating_sets))
        if len(locating_sets) != 0:
            min_len = len(A)
            for l in locating_sets:
                if len(l) < min_len:
                    min_len = len(l)
            print('The locating number is '+str(min_len))
        if len(locating_sets) != 0:
            break
```

Sample computation using Sage

Sample computation using Sage

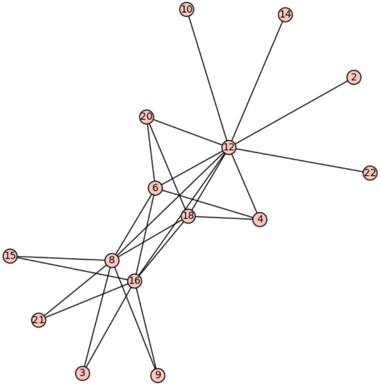


Sample computation using Sage



$loc(G) = 9$.
Minimal locating set: $\{2, 3, 4, 6, 8, 9, 10, 14, 15\}$

Sample computation using Sage



$loc(G) = 9$.
Minimal locating set: $\{2, 3, 4, 6, 8, 9, 10, 14, 15\}$
Computation time in Sage: ≈ 25 sec.

Exploring locating sets of disconnected graphs

Exploring locating sets of disconnected graphs

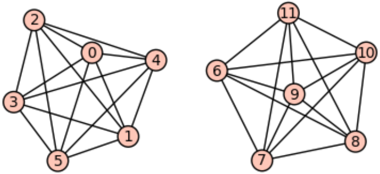


Figure: A disconnected graph G with two disjoint complete subgraphs G_1 and G_2.

$loc(G) = 10$

Exploring locating sets of disconnected graphs

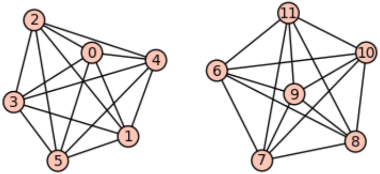


Figure: A disconnected graph G with two disjoint complete subgraphs G_1 and G_2.

$$loc(G) = 10 = 5 + 5$$

Exploring locating sets of disconnected graphs

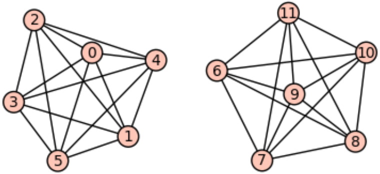


Figure: A disconnected graph G with two disjoint complete subgraphs G_1 and G_2.

$$loc(G) = 10 = 5 + 5 = loc(G_1) + loc(G_2)$$

Exploring locating sets of disconnected graphs

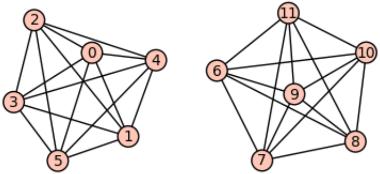


Figure: A disconnected graph G with two disjoint complete subgraphs G_1 and G_2.

$loc(G) = 10 = 5 + 5 = loc(G_1) + loc(G_2)$
$W = \{0, 1, 2, 3, 4, 6, 7, 8, 9, 10\}$

Exploring locating sets of disconnected graphs

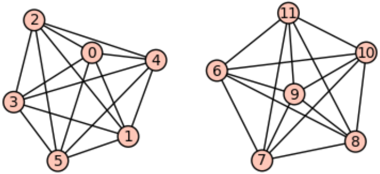


Figure: A disconnected graph G with two disjoint complete subgraphs G_1 and G_2.

$$loc(G) = 10 = 5 + 5 = loc(G_1) + loc(G_2)$$
$$W = \{0, 1, 2, 3, 4, 6, 7, 8, 9, 10\} = \{0, 1, 2, 3, 4\} \cup \{6, 7, 8, 9, 10\}$$

Minimal locating sets for disconnected graphs

Theorem

Let G be a disconnected graph that consists of k components G_i where $|G_i| \geq 2$.

Minimal locating sets for disconnected graphs

Theorem

Let G be a disconnected graph that consists of k components G_i where $|G_i| \geq 2$.

Then W is a locating set for G if and only if for each G_i there exists an $W_i \subseteq W$ such that W_i is a locating set for G_i.

Minimal locating sets for disconnected graphs

Proof

($\Rightarrow$) Let W be a locating set G.

Minimal locating sets for disconnected graphs

Proof

($\Rightarrow$) Let W be a locating set G. We will show $W \cap A_i \subseteq W$ is a locating set for A_i.

Minimal locating sets for disconnected graphs

Proof

($\Rightarrow$) Let W be a locating set G. We will show $W \cap A_i \subseteq W$ is a locating set for A_i. For $a, b \in A_i$, we have $c_W(a) \neq c_W(b)$ because W is a locating set for G.

Minimal locating sets for disconnected graphs

Proof

($\Rightarrow$) Let W be a locating set G. We will show $W \cap A_i \subseteq W$ is a locating set for A_i. For $a, b \in A_i$, we have $c_W(a) \neq c_W(b)$ because W is a locating set for G. Thus, $c_{W \cap A_i}(a) \neq c_{W \cap A_i}(b)$, which gives that $W \cap A_i$ is a locating set for A_i.

Minimal locating sets for disconnected graphs

Proof

($\Leftarrow$) We will show that $W = \bigcup_{i=1}^{k}(W \cap A_i)$ is a locating set for G.

Minimal locating sets for disconnected graphs

Proof

($\Leftarrow$) We will show that $W = \bigcup_{i=1}^{k}(W \cap A_i)$ is a locating set for G.
Let $a \in A_i, b \in A_j$ and suppose $i = j$.

Minimal locating sets for disconnected graphs

Proof

($\Leftarrow$) We will show that $W = \bigcup_{i=1}^{k}(W \cap A_i)$ is a locating set for G. Let $a \in A_i, b \in A_j$ and suppose $i = j$. We have $c_{W\cap A_i}(a) \neq c_{W\cap A_i}(b)$ because $W \cap A_i$ is a locating set for A_i.

Minimal locating sets for disconnected graphs

Proof

($\Leftarrow$) We will show that $W = \bigcup_{i=1}^{k}(W \cap A_i)$ is a locating set for G.
Let $a \in A_i, b \in A_j$ and suppose $i = j$. We have
$c_{W \cap A_i}(a) \neq c_{W \cap A_i}(b)$ because $W \cap A_i$ is a locating set for A_i.
Since $W \cap A_i \subseteq W$, we have $c_W(a) \neq c_W(b)$.

Minimal locating sets for disconnected graphs

Proof

($\Leftarrow$) We will show that $W = \bigcup_{i=1}^{k} (W \cap A_i)$ is a locating set for G. Let $a \in A_i, b \in A_j$ and suppose $i = j$. We have $c_{W \cap A_i}(a) \neq c_{W \cap A_i}(b)$ because $W \cap A_i$ is a locating set for A_i. Since $W \cap A_i \subseteq W$, we have $c_W(a) \neq c_W(b)$. Suppose $i \neq j$.

Minimal locating sets for disconnected graphs

Proof

($\Leftarrow$) We will show that $W = \bigcup_{i=1}^{k}(W \cap A_i)$ is a locating set for G. Let $a \in A_i, b \in A_j$ and suppose $i = j$. We have $c_{W \cap A_i}(a) \neq c_{W \cap A_i}(b)$ because $W \cap A_i$ is a locating set for A_i. Since $W \cap A_i \subseteq W$, we have $c_W(a) \neq c_W(b)$. Suppose $i \neq j$. Without loss of generality, let $c \in W \cap A_i$.

Minimal locating sets for disconnected graphs

Proof

($\Leftarrow$) We will show that $W = \bigcup_{i=1}^{k}(W \cap A_i)$ is a locating set for G. Let $a \in A_i, b \in A_j$ and suppose $i = j$. We have $c_{W \cap A_i}(a) \neq c_{W \cap A_i}(b)$ because $W \cap A_i$ is a locating set for A_i. Since $W \cap A_i \subseteq W$, we have $c_W(a) \neq c_W(b)$. Suppose $i \neq j$. Without loss of generality, let $c \in W \cap A_i$. This gives $d(a, c) = n$ for some $n \in \mathbb{N}$ so $d(b, c) = \infty$.

Minimal locating sets for disconnected graphs

Proof

($\Leftarrow$) We will show that $W = \bigcup_{i=1}^{k}(W \cap A_i)$ is a locating set for G. Let $a \in A_i, b \in A_j$ and suppose $i = j$. We have $c_{W \cap A_i}(a) \neq c_{W \cap A_i}(b)$ because $W \cap A_i$ is a locating set for A_i. Since $W \cap A_i \subseteq W$, we have $c_W(a) \neq c_W(b)$. Suppose $i \neq j$. Without loss of generality, let $c \in W \cap A_i$. This gives $d(a, c) = n$ for some $n \in \mathbb{N}$ so $d(b, c) = \infty$. Hence, $c_W(a) \neq c_W(b)$ and W is a locating set for G.

Locating numbers for disconnected graphs

Result

Let G be a disconnected graph consisting of k components $G_1, G_2, \ldots, G_k$ such that $|G_i| \geq 2$ and $loc(G_i) = a_i$.

Locating numbers for disconnected graphs

Result

Let G be a disconnected graph consisting of k components $G_1, G_2, \ldots, G_k$ such that $|G_i| \geq 2$ and $loc(G_i) = a_i$.

Then $loc(G) = \sum_{i=1}^{k} a_i$.

Locating numbers for disconnected graphs

Result

Let G be a disconnected graph consisting of $n > 1$ isolated vertices.

Locating numbers for disconnected graphs

Result

Let G be a disconnected graph consisting of $n > 1$ isolated vertices.

Then $loc(G) = n - 1$.

Locating numbers for disconnected graphs

Proof

Let $W = \{v_1, v_2, \ldots, v_{n-1}\}$ where $v_i \in G$.

Locating numbers for disconnected graphs

Proof

Let $W = \{v_1, v_2, \ldots, v_{n-1}\}$ where $v_i \in G$. We will show that W is a locating set and that removing any vertices from W will result in W no longer being a locating set.

Locating numbers for disconnected graphs

Proof

Let $W = \{v_1, v_2, \ldots, v_{n-1}\}$ where $v_i \in G$. We will show that W is a locating set and that removing any vertices from W will result in W no longer being a locating set.
Consider $v_j \in W$ where $1 \leq j \leq n-1$.

Locating numbers for disconnected graphs

Proof

Let $W = \{v_1, v_2, \ldots, v_{n-1}\}$ where $v_i \in G$. We will show that W is a locating set and that removing any vertices from W will result in W no longer being a locating set.
Consider $v_j \in W$ where $1 \leq j \leq n-1$. The corresponding locating code $c_W(v_j)$ will have 0 in the j^{th} component because $d(v_j, v_j) = 0$.

Locating numbers for disconnected graphs

Proof

Let $W = \{v_1, v_2, \ldots, v_{n-1}\}$ where $v_i \in G$. We will show that W is a locating set and that removing any vertices from W will result in W no longer being a locating set.
Consider $v_j \in W$ where $1 \leq j \leq n-1$. The corresponding locating code $c_W(v_j)$ will have 0 in the j^{th} component because $d(v_j, v_j) = 0$. Similarly, the j^{th} component will not be 0 for all other locating codes with respect to W.

Locating numbers for disconnected graphs

Proof

Let $W = \{v_1, v_2, \ldots, v_{n-1}\}$ where $v_i \in G$. We will show that W is a locating set and that removing any vertices from W will result in W no longer being a locating set.

Consider $v_j \in W$ where $1 \leq j \leq n-1$. The corresponding locating code $c_W(v_j)$ will have 0 in the j^{th} component because $d(v_j, v_j) = 0$. Similarly, the j^{th} component will not be 0 for all other locating codes with respect to W. Hence, $c_W(v_j)$ is distinct.

Locating numbers for disconnected graphs

Proof

Let $W = \{v_1, v_2, \ldots, v_{n-1}\}$ where $v_i \in G$. We will show that W is a locating set and that removing any vertices from W will result in W no longer being a locating set.

Consider $v_j \in W$ where $1 \leq j \leq n-1$. The corresponding locating code $c_W(v_j)$ will have 0 in the j^{th} component because $d(v_j, v_j) = 0$. Similarly, the j^{th} component will not be 0 for all other locating codes with respect to W. Hence, $c_W(v_j)$ is distinct.

Locating numbers for disconnected graphs

Proof

Now consider $v_n \in G$ where $v_n \neq v_i$ for $1 \leq i \leq n-1$.

Locating numbers for disconnected graphs

Proof

Now consider $v_n \in G$ where $v_n \neq v_i$ for $1 \leq i \leq n-1$. The corresponding locating code $c_W(v_n)$ has ∞ in every component.

Locating numbers for disconnected graphs

Proof

Now consider $v_n \in G$ where $v_n \neq v_i$ for $1 \leq i \leq n-1$. The corresponding locating code $c_W(v_n)$ has ∞ in every component. Since this is not the case for any other vertices in G, we have that $c_W(v_n)$ is distinct.

Locating numbers for disconnected graphs

Proof

Now consider $v_n \in G$ where $v_n \neq v_i$ for $1 \leq i \leq n-1$. The corresponding locating code $c_W(v_n)$ has ∞ in every component. Since this is not the case for any other vertices in G, we have that $c_W(v_n)$ is distinct. Therefore, all locating codes with respect to W are unique so W is a locating set.

Locating numbers for disconnected graphs

Proof

Now consider $v_n \in G$ where $v_n \neq v_i$ for $1 \leq i \leq n-1$. The corresponding locating code $c_W(v_n)$ has ∞ in every component. Since this is not the case for any other vertices in G, we have that $c_W(v_n)$ is distinct. Therefore, all locating codes with respect to W are unique so W is a locating set. Consider $W - \{v_i\}$ for some $v_i \in W$. Then $c_W(v_i) = c_W(v_n) = (\infty, \infty \ldots, \infty)$, so $W - \{v_i\}$ is not a locating set.

Locating numbers for disconnected graphs

Proof

Now consider $v_n \in G$ where $v_n \neq v_i$ for $1 \leq i \leq n-1$. The corresponding locating code $c_W(v_n)$ has ∞ in every component. Since this is not the case for any other vertices in G, we have that $c_W(v_n)$ is distinct. Therefore, all locating codes with respect to W are unique so W is a locating set. Consider $W - \{v_i\}$ for some $v_i \in W$. Then $c_W(v_i) = c_W(v_n) = (\infty, \infty \ldots, \infty)$, so $W - \{v_i\}$ is not a locating set. Hence, W is a minimal locating set and $loc(G) = n - 1$.

Locating numbers for disconnected graphs

Theorem

Let G be a disconnected graph such that $G = (\bigcup_{i=1}^{k} G_i) \cup \{v_i, v_2, \ldots, v_n\}$ where G_i is a connected subgraph two or more vertices and v_j is an isolated vertex.

Locating numbers for disconnected graphs

Theorem

Let G be a disconnected graph such that $G = (\bigcup_{i=1}^{k} G_i) \cup \{v_i, v_2, \ldots, v_n\}$ where G_i is a connected subgraph two or more vertices and v_j is an isolated vertex.

Then $loc(G) = \sum_{i=1}^{k} loc(G_i) + (n-1)$

Generalized characterization results

Generalized characterization results

Result

Let G be a graph on n vertices.

Generalized characterization results

Result

Let G be a graph on n vertices.

Then $loc(G) = n - 1$ if and only if G is complete or G consists of only isolated vertices.

Generalized characterization results

Proof

($\Leftarrow$) If $G \cong K_n$, then $loc(G) = n - 1$ as shown by Pirzada et. al.

Generalized characterization results

Proof

($\Leftarrow$) If $G \cong K_n$, then $loc(G) = n - 1$ as shown by Pirzada et. al. If G consists of n isolated vertices, then $loc(G) = n - 1$ by the previous Lemma.

Generalized characterization results

Proof

($\Leftarrow$) If $G \cong K_n$, then $loc(G) = n - 1$ as shown by Pirzada et. al. If G consists of n isolated vertices, then $loc(G) = n - 1$ by the previous Lemma.

Generalized characterization results

Proof

($\Rightarrow$) Suppose $loc(G) = n - 1$.

Generalized characterization results

Proof

($\Rightarrow$) Suppose $loc(G) = n - 1$.

Case 1: If G is connected, then $G \cong K_n$.

Generalized characterization results

Proof

Case 2: Suppose A_i is the only connected component with $|A_i| \geq 2$ and each $v_k \notin A_i$ is isolated.

Generalized characterization results

Proof

Case 2: Suppose A_i is the only connected component with $|A_i| \geq 2$ and each $v_k \notin A_i$ is isolated. Let $|A_i| = x$. Then there are $n - x$ isolated vertices in G.

Generalized characterization results

Proof

Case 2: Suppose A_i is the only connected component with $|A_i| \geq 2$ and each $v_k \notin A_i$ is isolated. Let $|A_i| = x$. Then there are $n - x$ isolated vertices in G. By our previous Theorem, we have $loc(G) = loc(A_i) + (n - x - 1)$.

Generalized characterization results

Proof

Case 2: Suppose A_i is the only connected component with $|A_i| \geq 2$ and each $v_k \notin A_i$ is isolated. Let $|A_i| = x$. Then there are $n - x$ isolated vertices in G. By our previous Theorem, we have $loc(G) = loc(A_i) + (n - x - 1)$. Since $loc(A_i) < |A_i|$, we have $loc(G) < n - 1$.

Generalized characterization results

Proof

Case 2: Suppose A_i is the only connected component with $|A_i| \geq 2$ and each $v_k \notin A_i$ is isolated. Let $|A_i| = x$. Then there are $n - x$ isolated vertices in G. By our previous Theorem, we have $loc(G) = loc(A_i) + (n - x - 1)$. Since $loc(A_i) < |A_i|$, we have $loc(G) < n - 1$. Thus, there do not exist any disjoint components in G.

Generalized characterization results

Proof

Case 2: Suppose A_i is the only connected component with $|A_i| \geq 2$ and each $v_k \notin A_i$ is isolated. Let $|A_i| = x$. Then there are $n - x$ isolated vertices in G. By our previous Theorem, we have $loc(G) = loc(A_i) + (n - x - 1)$. Since $loc(A_i) < |A_i|$, we have $loc(G) < n - 1$. Thus, there do not exist any disjoint components in G. Therefore, G consists entirely of isolated vertices.

Generalized characterization results

Proof

Case 3: Suppose G is disconnected and there exist at least two disjoint components $A_i, A_j \subseteq G$.

Generalized characterization results

Proof

Case 3: Suppose G is disconnected and there exist at least two disjoint components $A_i, A_j \subseteq G$. From Pirzada et. al., we have $loc(A_i) \leq |A_i| - 1$ and $loc(A_j) \leq |A_j| - 1$.

Generalized characterization results

Proof

Case 3: Suppose G is disconnected and there exist at least two disjoint components $A_i, A_j \subseteq G$. From Pirzada et. al., we have $loc(A_i) \leq |A_i| - 1$ and $loc(A_j) \leq |A_j| - 1$. Thus, we have that $loc(G) \leq n - 2 < n - 1$.

Generalized characterization results

Proof

Case 3: Suppose G is disconnected and there exist at least two disjoint components $A_i, A_j \subseteq G$. From Pirzada et. al., we have $loc(A_i) \leq |A_i| - 1$ and $loc(A_j) \leq |A_j| - 1$. Thus, we have that $loc(G) \leq n - 2 < n - 1$. Hence, there cannot be two or more disjoint components in G.

Generalized characterization results

Proof

Case 3: Suppose G is disconnected and there exist at least two disjoint components $A_i, A_j \subseteq G$. From Pirzada et. al., we have $loc(A_i) \leq |A_i| - 1$ and $loc(A_j) \leq |A_j| - 1$. Thus, we have that $loc(G) \leq n - 2 < n - 1$. Hence, there cannot be two or more disjoint components in G.

Generalized characterization results

Generalized characterization results

Result

Let G be a graph. Then $loc(G) = 1$ if and only if G is isomorphic to one of the following:

Generalized characterization results

Result

Let G be a graph. Then $loc(G) = 1$ if and only if G is isomorphic to one of the following:

(i) P_n
(ii) $P_n \cup \{v_1\}$
(iii) $\{v_1, v_2\}$

where P_n is a path on n vertices and v_i is an isolated vertex.

Generalized characterization results

Result

Let G be a graph. Then $loc(G) = 2$ if and only if G is isomorphic to one of the following:

Generalized characterization results

Result

Let G be a graph. Then $loc(G) = 2$ if and only if G is isomorphic to one of the following:

(i) C_n
(ii) $C_n \cup \{v_1\}$
(iii) $P_n \cup P_m$
(iv) $P_n \cup P_m \cup \{v_1\}$
(v) $P_n \cup \{v_1, v_2\}$
(vi) $\{v_1, v_2, v_3\}$

where P_n is a path on n vertices, C_m is a cycle on m vertices, and v_i is an isolated vertex.

Future work

Future work

- Graph sums and graph complements

Future work

- Graph sums and graph complements
- Graphs with cut-points and bridges

Future work

- Graph sums and graph complements
- Graphs with cut-points and bridges
- Application to graphical representations of commutative rings

Future work

- Graph sums and graph complements
- Graphs with cut-points and bridges
- Application to graphical representations of commutative rings
- Directed Graphs

Acknowledgements

Acknowledgements

- Dr. Joe Stickles

Acknowledgements

- Dr. Joe Stickles
- Millikin University Department of Mathematics

Acknowledgements

- Dr. Joe Stickles
- Millikin University Department of Mathematics
- Illinois College

References

S. Pirzada, R. Raja, S. Redmond, Locating Sets and Numbers of Graphs Associated to Commutative Rings, *Journal of Algebra and Its Applications* **13** (2013), 1-17.

S. Pirzada, R. Raja, S. Redmond, On Locating Numbers and Codes of Zero Divisor Graphs Associated with Commutative Rings, preprint.

Thank you!

Questions?