Solving graph isomorphism problem in polynomial time

Yasunori Ohto

Abstract

We show that the graph isomorphism problem is solvable in polynomial time. First, we prove the theorem to obtain the automorphisms of $G$ using eigenvalue sets. Next, we construct an algorithm to determine whether two given graphs $G_a$ and $G_b$ are isomorphic using this result. The computational complexity to detect whether the two graphs are isomorphic is $O(n^6)$.

Index Terms

graph isomorphism problem, graph spectrum, polynomial time computation.

I. INTRODUCTION

The graph isomorphism problem [1] is to determine whether two given graphs are isomorphic. This problem is one of the major problems in theoretical computer science, especially regarding the class of its computational complexity [2]. There are practical algorithms that can determine whether two graphs are isomorphic [3], [4], [5]. These methods can obtain correct results at a practical level. On the theoretical side, a quasi-polynomial algorithm has been proposed [6], [7]. On the other hand, polynomial-time isomorphism detection algorithms exist for special graphs [8], [9], [10].

The set of the eigenvalues of the adjacency matrix of a graph indicates the characteristics of the graph. However, if the two sets of eigenvalues are the same, such graphs are called cospectral graphs [11], and the graphs might not be isomorphic. Therefore, we cannot determine the isomorphism of two graphs by whether their eigenvalue sets are only the same. When there two sets are the same, if we restrict the multiplicities of the eigenvalues, we can determine whether the graphs are isomorphic in polynomial time [12], [13]. However, it is unclear whether a polynomial-time algorithm exists for general graphs.

In this paper, we show that the graph isomorphism problem is solvable in polynomial time. First, we define the following functions. Let $S$ be a vertex-weighted graph. Let $V_{w0}(S)$ be the set of vertices of $S$ with weight 0. Let $S_g(S, v, w)$ be the vertex-weighted graph in which weight $w \in \mathbb{N}$ is given to vertex $v$ of $S$. Let $Ev(S)$ be the set of eigenvalues of the adjacency matrix of $S$. Next, we prove Theorem I.1 to obtain the automorphisms of $S$ using eigenvalue sets.

**Theorem I.1.** Let $S_{v_i} = S_g(S, v_i, w)$ and $S_{v_j} = S_g(S, v_j, w)$ with $v_i, v_j \in V_{w0}(S)$, $v_i \neq v_j$ and $w > 0$. When $Ev(S_{v_i}) = Ev(S_{v_j})$, $S_{v_i}$ and $S_{v_j}$ are isomorphic.

Next, we construct an algorithm to determine whether two given graphs $G_a$ and $G_b$ are isomorphic using this result. Write $n$ for the number of vertices of these graphs. Let $S_{a_0} = G_a$ and $S_{b_0} = G_b$. Consider a vertex weight $w_i > 0 \notin \{w_j|0 \leq j < i\}$. Let $S_{a_{i+1}}$ be $S_g(S_{a_i}, v_{a_i}, w_i)$ with $v_{a_i} \in V_{w0}(S_{a_i})$. Let $S_{b_{i+1}}$ be $S_g(S_{b_i}, v_{b_i}, w_i)$ with $v_{b_i} \in V_{w0}(S_{b_i})$. Let $Ev(S_{a_{i+1}}) = Ev(S_{b_{i+1}})$. Then, we check the vertex mapping $\{v_{a_i} \mapsto v_{b_i}|0 \leq i < n\}$ to determine whether $G_a$ and $G_b$ are isomorphic. Note that we compare the two eigenvalue sets without real number calculations by using Frobenius normal form [14], [15]. The computational complexity for determining whether the two graphs are isomorphic is $O(n^6)$, where $n$ represents the number of vertices in the graph.

This paper is organized as follows. Section II provides the proofs used to determine whether two graphs are isomorphic. Section III presents an algorithm to solve this problem. Finally, Section IV presents a conclusion regarding the result of this paper.
II. PROOF

In this section, we provide the proofs for the results of the determination of whether two given graphs are isomorphic.

A. Preparation

We define the following functions, which will be used in the proofs and the methods. Suppose \( S \) is a vertex-weighted graph. Let \( V_{w0}(S) \) be the set of vertices of \( S \) with weight 0. Let \( Sg(S,v,w) \) be the vertex-weighted graph in which the weight \( w \in \mathbb{N} \) is given to vertex \( v \) of \( S \). Denote the adjacency matrix of \( S \) by \( A(S) \). Let \( Ev(S) \) be the set (with multiplicities) of eigenvalues of \( A(S) \).

B. Obtaining the automorphisms

Theorem I.1 and Corollary II.1 prove that it is possible to obtain the automorphisms of \( S \) using eigenvalue sets.

**Theorem I.1.** Let \( S_{v_1} = Sg(S,v_1,w) \) and \( S_{v_j} = Sg(S,v_j,w) \) with \( v_1, v_j \in V_{w0}(S) \), \( v_1 \neq v_j \) and \( w > 0 \). When \( Ev(S_{v_1}) = Ev(S_{v_j}) \), \( S_{v_1} \) and \( S_{v_j} \) are isomorphic.

**Proof.** We show that if \( Ev(S_{v_1}) = Ev(S_{v_j}) \), then \( S_{v_1} \) and \( S_{v_j} \) are not cospectral but isomorphic.

Let \( A(S_{v_1}) \) and \( A(S_{v_j}) \) be \( A_{v_1} \) and \( A_{v_j} \), respectively. When there exists a permutation matrix \( P \) such that \( A_{v_1} = P^tA_{v_j}P \), \( S_{v_1} \) and \( S_{v_j} \) are isomorphic. Denote the eigenfunctions of \( A_{v_1} \) and \( A_{v_j} \) by \( f_{v_1} \) and \( f_{v_j} \), respectively. When \( f_{v_1} \) and \( f_{v_j} \) are the same, the eigenvalue sets of \( A_{v_1} \) and \( A_{v_j} \) are the same. Therefore, we will prove that such a nontrivial permutation matrix exists when \( f_{v_1} - f_{v_j} = 0 \).

Without loss of generality, we may assume \( i = 1 \) and \( j = 2 \). We show the characteristic polynomials \( f_{v_1} \) and \( f_{v_2} \) as below.

\[
\begin{align*}
 f_{v_1} &= |A_{v_1} - \lambda I| \\
 &= \begin{vmatrix}
 w - \lambda & a_{1,2} & a_{1,3} & a_{1,4} & \cdots & a_{1,n} \\
 a_{2,1} & -\lambda & a_{2,3} & a_{2,4} & \cdots & a_{2,n} \\
 a_{3,1} & a_{3,2} & w - \lambda & a_{3,4} & \cdots & a_{3,n} \\
 a_{4,1} & a_{4,2} & a_{4,3} & \cdots & \cdots & \cdots \\
 \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \\
 a_{n,1} & a_{n,2} & a_{n,3} & \cdots & \cdots & w_n - \lambda
\end{vmatrix}, \\
 f_{v_2} &= |A_{v_2} - \lambda I| \\
 &= \begin{vmatrix}
 -\lambda & a_{1,2} & a_{1,3} & a_{1,4} & \cdots & a_{1,n} \\
 a_{2,1} & w - \lambda & a_{2,3} & a_{2,4} & \cdots & a_{2,n} \\
 a_{3,1} & a_{3,2} & w - \lambda & a_{3,4} & \cdots & a_{3,n} \\
 a_{4,1} & a_{4,2} & a_{4,3} & \cdots & \cdots & \cdots \\
 \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \\
 a_{n,1} & a_{n,2} & a_{n,3} & \cdots & \cdots & w_n - \lambda
\end{vmatrix}.
\end{align*}
\]

The weights of the vertices are \( w, w_3, \ldots, w_n \), all of which are integers. Then,

\[
 f_{v_1} - f_{v_2} = w \begin{vmatrix}
 0 & a_{2,3} & a_{2,4} & \cdots & a_{2,n} \\
 a_{3,2} & w - \lambda & a_{3,4} & \cdots & a_{3,n} \\
 a_{4,2} & a_{4,3} & \cdots & a_{4,n} \\
 \vdots & \vdots & \ddots & \ddots & \ddots \\
 a_{n,2} & a_{n,3} & \cdots & \cdots & w_n - \lambda
\end{vmatrix} - w \begin{vmatrix}
 0 & a_{1,3} & a_{1,4} & \cdots & a_{1,n} \\
 a_{3,1} & w - \lambda & a_{3,4} & \cdots & a_{3,n} \\
 a_{4,1} & a_{4,3} & \cdots & a_{4,n} \\
 \vdots & \vdots & \ddots & \ddots & \ddots \\
 a_{n,1} & a_{n,3} & \cdots & \cdots & w_n - \lambda
\end{vmatrix} = 0.
\]
If $n = 2$, $f_{v_1}$ and $f_{v_2}$ are the same. Hence, in this case, $S_{v_1}$ and $S_{v_2}$ are isomorphic. We treat the case of $n = 3$ as follows. Equation 1 becomes

$$f_{v_1} - f_{v_2} = w \begin{vmatrix} 0 & a_{2,3} \\ a_{3,2} & w_3 - \lambda \end{vmatrix} - w \begin{vmatrix} 0 & a_{1,3} \\ a_{3,1} & w - \lambda \end{vmatrix}$$

$$= w(a_{2,3}a_{3,2} - a_{1,3}a_{3,1})$$

So, when $a_{2,3} = a_{1,3}$, $f_{v_1}$ and $f_{v_2}$ are the same. For this case, then, $S_{v_1}$ and $S_{v_2}$ are isomorphic.

Let $n > 3$. Suppose the matrix $A'$ is as follows.

$$A' = \begin{pmatrix} a_{3,3} & \cdots & a_{3,n} \\ a_{4,3} & \cdots & a_{4,n} \\ \vdots & \vdots & \vdots \\ a_{n,3} & \cdots & a_{n,n} \end{pmatrix}$$

Let vertex $u_1 = (a_{1,3}, a_{1,4}, \ldots, a_{1,n})^t$ and $u_2 = (a_{2,3}, a_{2,4}, \ldots, a_{2,n})^t$. Then, Equation 1 becomes as follows.

$$f_{v_1} - f_{v_2} = w \begin{vmatrix} 0 & u_{1,2}^t \\ u_{2,2} & A' - \lambda I \end{vmatrix} - w \begin{vmatrix} 0 & u_1^t \\ u_2 & A' - \lambda I \end{vmatrix}$$

$$= 0.$$ 

In order for $f_{v_1}$ and $f_{v_2}$ to be the same, it is necessary that $f_{v_1} - f_{v_2} = 0$ for all $\lambda$. So, we assume $|A' - \lambda I| \neq 0$. Then,

$$f_{v_1} - f_{v_2} = \begin{vmatrix} |A' - \lambda I||0 - u_{1,2}^t(A' - \lambda I)^{-1}u_2| \\ - |A' - \lambda I||0 - u_{1,2}^t(A' - \lambda I)^{-1}u_1| \\ w|A' - \lambda I|(u_2 - u_1)^t(A' - \lambda I)^{-1}(u_2 - u_1) \\ - w|A' - \lambda I|(u_2 - u_1)^t(A' - \lambda I)^{-1}(u_2 - u_1) \end{vmatrix}$$

$$= 0.$$ 

When $u_1 = u_2$, $f_{v_1}$ and $f_{v_2}$ are the same. In this case, then, $S_{v_1}$ and $S_{v_2}$ are isomorphic.

Let $u_2 \neq u_1$. When $(u_2 - u_1)^t(A' - \lambda I)^{-1}(u_2 - u_1) = 0$, $u_2 - u_1$ and $(A' - \lambda I)^{-1}(u_2 - u_1)$ are orthogonal. So,

$$(u_2 - u_1)^t(A' - \lambda I)(u_2 - u_1) = u_{1,2}^tA'u_2 - u_{1,2}^tA'u_1 - u_{2,2}^t\lambda Iu_2 + u_{2,2}^t\lambda Iu_1$$

$$= 0.$$ 

In order for $f_{v_1}$ and $f_{v_2}$ to be the same, it is necessary that $f_{v_1} - f_{v_2} = 0$ for all $\lambda$. So, the number of elements with value 1 in $u_2$ and $u_1$ is the same.

Since $u_2 - u_1$ and $(A' - \lambda I)(u_2 - u_1)$ are orthogonal,

$$(u_2 - u_1)^tA'(u_2 - u_1) = (u_2 - u_1)^tP^tA'P'(u_2 - u_1)$$

$$= (u_2 - u_1)^tP^tA'P'(u_2 - u_1)$$

$$= 0$$

with $P'$ a liner operator. When $A_1$ and $A_2$ have the same eigenvalue set, there exists a set of nontrivial permutation matrices $\{P'|P^tA'P' = A' \land (u_2 - u_1) = P'(u_2 - u_1)\}$. So, $S_{v_1}$ and $S_{v_2}$ are isomorphic.

**Corollary II.1.** Let $S_{v_1} = Sg(S, v_i, w)$ and $S_{v_j} = Sg(S, v_j, w)$ with $v_i, v_j \in V_{w0}(S)$, $v_i \neq v_j$ and $w > 0$. If $Ev(S_{v_1}) \neq Ev(S_{v_j})$, then $S_{v_1}$ and $S_{v_j}$ are not isomorphic.

**Proof.** By applying a permutation matrix $P$, $P^tA(S_{v_1})P$ is not equal to $A(S_{v_1})$. So, there is no bijection between $S_{v_1}$ and $S_{v_j}$. Therefore, $S_{v_1}$ and $S_{v_j}$ are not isomorphic. 

C. Obtaining vertex mapping and detecting whether two graphs are isomorphic

Corollaries II.2 and II.3 prove that it is possible to obtain vertex mapping and detect whether two graphs are isomorphic.

Isomorphism does not exist when the same eigenvalue set does not exist.

**Corollary II.2.** Suppose $S_{a_i}$ and $S_{b_i}$ are vertex-weighted graphs. Let $Ev(S_{a_i}) = Ev(S_{b_i})$. Let $v_{a_i} \in V_{w0}(S_{a_i})$ and $w > 0$. When $Ev(Sg(S_{a_i}, v_{a_i}, w)) \neq Ev(Sg(S_{b_i}, v_{b_i}, w))$ for any $v_{b_i} \in V_{w0}(S_{b_i})$, $S_{a_i}$ and $S_{b_i}$ are not isomorphic.

**Proof:** By applying a permutation matrix $P$, $P^tA(Sg(S_{b_i}, v_{b_i}, w))P$ is not equal to $A(Sg(S_{a_i}, v_{a_i}, w))$ for any $v_{a_i} \in V_{w0}(S_{a_i})$ and $v_{b_i} \in V_{w0}(S_{b_i})$. Therefore, $S_{a_i}$ and $S_{b_i}$ are isomorphic.

When isomorphism exists, this relationship keeps even if the weighting of vertices is removed.

**Corollary II.3.** Suppose $S_{a_i}$ and $S_{b_i}$ are vertex-weighted graphs. Let $Ev(S_{a_i}) = Ev(S_{b_i})$ and $w > 0$. Let $S_{a_{i+1}} = Sg(S_{a_i}, v_{a_i}, w)$ with $v_{a_i} \in V_{w0}(S_{a_i})$. Let $S_{b_{i+1}} = Sg(S_{b_i}, v_{b_i}, w)$ with $v_{b_i} \in V_{w0}(S_{b_i})$. Let $Ev(S_{a_{i+1}}) = Ev(S_{b_{i+1}})$. Now, $S_{a_{i+1}}$ and $S_{b_{i+1}}$ are isomorphic if, and only if, $S_{a_i}$ and $S_{b_i}$ are isomorphic.

**Proof:** Let $V_a$ be the set of vertices of $S_{a_{i+1}}$ and $V_b$ that of $S_{b_{i+1}}$. When $S_{a_{i+1}}$ and $S_{b_{i+1}}$ are not isomorphic, there is no bijection $V_a \rightarrow V_b$.

When $S_{a_{i+1}}$ and $S_{b_{i+1}}$ are isomorphic, there exists a bijection $V_a \rightarrow V_b$. From Theorem I.1 and Corollary II.1, for any $v_{a_i}$ and $v_{b_i}$ such that $Ev(S_{a_{i+1}}) = Ev(S_{b_{i+1}})$, $S_{a_i}$ and $S_{b_i}$ are isomorphic.

### III. Algorithm

In this section, we present a polynomial-time algorithm to determine whether two graphs $G_a$ and $G_b$ are isomorphic. We assume that the number of vertices of the graphs is $n$.

**A. Comparing the sets of eigenvalues**

This subsection shows that how to compare the sets of eigenvalues without real number calculations.

Since the elements of an adjacency matrix of a vertex-weighted graph are all integers, the coefficients of the eigenequation of this matrix are all integers. We calculate the Frobenius normal form to obtain the coefficients of the eigenequation of the adjacency matrix of a vertex-weighted graph without real number calculations. Then, we compare the coefficients to determine whether the eigenvalue sets are the same. The amount of computation required to convert an adjacency matrix into the Frobenius normal form is $O(n^4)$. The amount of computation to compare the coefficients of the two characteristic equations is $O(n)$.

**B. Determining whether two graphs are isomorphic**

This subsection determines whether two graphs $G_a$ and $G_b$ are isomorphic. We show the flow of the algorithm below.

Let $S_{a_0} = G_a$ and $S_{b_0} = G_b$. Consider a vertex weight $w_i > 0 \notin \{w_j | 0 \leq j < i\}$. Let $S_{a_{i+1}}$ be $Sg(S_{a_i}, v_{a_i}, w_i)$ with $v_{a_i} \in V_{w0}(S_{a_i})$. Let $S_{b_{i+1}}$ be $Sg(S_{b_i}, v_{b_i}, w_i)$ with $v_{b_i} \in V_{w0}(S_{b_i})$. Let $Ev(S_{a_{i+1}}) = Ev(S_{b_{i+1}})$. Then, we check the vertex mapping $\{v_{a_i} \mapsto v_{b_i} | 0 \leq i < n\}$ to determine whether $G_a$ and $G_b$ are isomorphic.

Function 1 determines whether two graphs $G_a$ and $G_b$ are isomorphic. From Theorem I.1 and Corollaries II.1, II.2, II.3, this function can detect whether two graphs are isomorphic. The amount of computation of this function is $O(n^6)$.

**Example III.1.** Figure 1 shows that graphs $G_a = (V_a, E_a)$ and $G_b = (V_b, E_b)$ as an example to determine graph isomorphism. At first, we clear hash $h$. Let $S_{a_0} = G_a$ and $S_{b_0} = G_b$. So, all vertices in $S_{a_0}$ and
Algorithm 1 Determining whether two graphs $G_a$ and $G_b$ are isomorphic.

1: function IS_ISOMORPHIC($G_a = (V_a, E_a)$, $G_b = (V_b, E_b)$)
2: $S_a \leftarrow G_a$ with all vertex weights equal to 0
3: $S_b \leftarrow G_b$ with all vertex weights equal to 0
4: if $Ev(S_a) \neq Ev(S_b)$ then
5:     return FALSE
6: end if
7: $w \leftarrow 2|V_a|$  
8: clear hash $h$
9: for each $v_a \in V_a$ do
10:     $\lambda \leftarrow Ev(Sg(S_a, v_a, w))$
11:     $v_b \leftarrow$ OBTAIN_VERTEX($S_b$, $\lambda$, $w$)
12:     if $v_b = \text{null}$ then
13:         return FALSE
14: end if
15: $S_a \leftarrow Sg(S_a, v_a, w)$
16: $S_b \leftarrow Sg(S_b, v_b, w)$
17: $w \leftarrow w + 2|V_a|$  
18: $h(v_a) \leftarrow v_b$
19: end for
20: return TEST_ISOMORPHIC($G_a$, $G_b$, $h$)
21: end function

Fig. 1. Graphs $G_a$ and $G_b$ as an example to determine graph isomorphism.

$S_{b_0}$ have weights of 0. Next, let $w_0 = 2|V_a|$. Let $S_{a_1} = Sg(S_{a_0}, p_1, w_0)$. Obtain the vertex $v_0 \in V_{a_0}(S_{b_0})$ such that $Ev(Sg(S_{b_0}, v_0, w_0)) = Ev(S_{a_1})$. Vertices $p_5, \ldots, p_8$ satisfy this condition. Since Theorem I.1 and Corollary II.1, we can select a vertex from any of them. So, we select $p_5$, then, let $S_{b_1} = Sg(S_{b_0}, p_5, w_0)$ and $h(p_1) = p_5$. Next, let $w_1 = w_0 + 2|V_a|$. Let $S_{a_2} = Sg(S_{a_1}, p_2, w_1)$. Obtain the vertex $v_1 \in V_{a_0}(S_{b_1})$ such that $Ev(Sg(S_{b_1}, v_1, w_1)) = Ev(S_{a_2})$. Vertices $p_6$ and $p_7$ satisfy this condition. We can select a vertex from any of them. So, we select $p_6$, then, let $S_{b_2} = Sg(S_{b_1}, p_6, w_1)$ and $h(p_2) = p_6$. Next, let $w_2 = w_1 + 2|V_a|$. Let $S_{a_3} = Sg(S_{a_2}, p_3, w_2)$. Obtain the vertex $v_2 \in V_{a_0}(S_{b_2})$ such that $Ev(Sg(S_{b_2}, v_2, w_2)) = Ev(S_{a_3})$. Vertex $p_8$ satisfies this condition. So, we use $p_8$, then, let $S_{b_3} = Sg(S_{b_2}, p_8, w_2)$ and $h(p_3) = p_8$. Next, let $w_3 = w_2 + 2|V_a|$. Let $S_{a_4} = Sg(S_{a_3}, p_4, w_3)$. Obtain the vertex $v_3 \in V_{a_0}(S_{b_3})$ such that $Ev(Sg(S_{b_3}, v_3, w_3)) = Ev(S_{a_4})$. Vertex $p_7$ satisfies this condition. So, we use $p_7$, then, let $S_{b_4} = Sg(S_{b_3}, p_7, w_3)$ and let $h(p_4) = p_7$. Finally, using the stack $h$ that stored the map between vertices of $G_a$ and $G_b$, we check whether a bijection exists between $G_a$ and $G_b$.

We show a function 2 to obtain a vertex $v$ such that $Ev(Sg(S, v, w)) = \lambda$ with $v \in V_{a_0}(S)$. So, the amount of computation of this function is $O(n^5)$.

Function 3 determines whether a map $v_a \mapsto v_b$ exists for the set of vertex pair $(v_a, v_b)$ of vertices
Algorithm 2 Obtaining a vertex $v$ such that $Ev(Sg(S, v, w)) = \lambda$ with $v \in V_{w0}(S)$.

```
1: function OBTAIN_VERTEX($S$, $\lambda$, $w$)
2:     for each $v \in V_{w0}(S)$ do
3:         if $Ev(Sg(S, v, w)) = \lambda$ then
4:             return $v$
5:     end if
6: end for
7: return null
8: end function
```

Algorithm 3 Determining whether a map $v_a \mapsto v_b$ exists for the set of vertex pair $(v_a, v_b)$ of vertices $v_a \in V_a$ and $v_b \in V_b$ of two graphs $G_a = (V_a, E_a)$ and $G_b = (V_b, E_b)$.

```
1: function TEST_ISOMORPHIC($G_a = (V_a, E_a)$, $G_b = (V_b, E_b)$, $h$)
2:     for $i \leftarrow 1$ to $|V_a| - 1$ do
3:         $v_i \leftarrow i$-th vertex in $V_a$
4:         for $j \leftarrow i + 1$ to $|V_a|$ do
5:             $v_j \leftarrow j$-th vertex in $V_a$
6:             if $(v_i, v_j) \in E_a$ then
7:                 if $(h(v_i), h(v_j)) \notin E_b$ then
8:                     return FALSE
9:                 end if
10:             else
11:                 if $(h(v_i), h(v_j)) \in E_b$ then
12:                     return FALSE
13:                 end if
14:             end if
15:         end for
16:     end for
17: return TRUE
18: end function
```

$v_a \in V_a$ and $v_b \in V_b$ of two graphs $G_a = (V_a, E_a)$ and $G_b = (V_b, E_b)$. The amount of computation of this function is $O(n^2)$.

IV. Conclusion

In this paper, we have presented an algorithm to detect whether two given graphs are isomorphic. It has polynomial time complexity. Note that this algorithm has a limitation in that it can only obtain one of the isomorphisms.

APPENDIX A

Definition

In this section, we give the definitions used in this paper.

Definition A.1. A graph $G = (V, E)$ is a pair consisting of a non-empty finite vertex set $V \neq \emptyset$ and an edge set $E$ that is a subset of $V^2$. The graph’s size is the number of its vertices $1 < n = |V|$. The number of vertices in a graph is assumed to be finite. In addition, we align the set $V$ with $\{v_1, \ldots, v_n\}$. There is an edge between vertices $v_a$ and $v_b$ when $(v_a, v_b)$ is an element of the set $E$. Also, edges have no direction. Moreover, the graph has no multiple edges between a pair of vertices, and there are no loops (i.e., $(v_a, v_a)$ is never an edge).
**Definition A.2.** A vertex-weighted graph $S = (V, E, z)$ is a graph with a function $z : V \rightarrow \mathbb{N}$ that gives the weights of the vertices. Then, a graph is a vertex-weighted graph in which the weights of all its vertices are 0.

**Definition A.3.** The adjacency matrix $A$ of a vertex-weighted graph $S = (V, E, z)$ with $n = |V|$ is an $n \times n$ symmetric matrix that is given as follows. The entries $a_{i,j}, v_i, v_j \in V, 0 < i, j \leq n$ of $A$ satisfy:

\[
\begin{cases}
(v_i, v_j) \in E & \text{if } a_{i,j} = a_{j,i} = 1, \\
(v_i, v_j) \notin E & \text{if } a_{i,j} = a_{j,i} = 0, \\
a_{i,i} = z(v_i).
\end{cases}
\]

**Definition A.4.** An isomorphism of graphs $G_a = (V_a, E_a)$ and $G_b = (V_b, E_b)$ is a bijection between the vertex sets of $G_a$ and $G_b$ $f : V_b \rightarrow V_a$ such that two vertices $v_i$ and $v_j$ of $G_a$ are adjacent in $G_a$ if and only if $f(v_a)$ and $f(v_b)$ are adjacent in $G_b$.

**References**


