Ucycles and Gucycles of Integer Compositions and Partitions

Grant Shirley Anant Godbole

East Tennessee State University

September 4, 2024

Introduction

A Universal Cycle, or Ucycle, is a non-repetitive cyclical representation of all values of a combinatorial object

Introduction

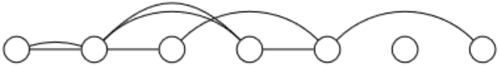
A Universal Cycle, or Ucycle, is a non-repetitive cyclical representation of all values of a combinatorial object; a DeBruijn Sequence is the most basic example, which is a sequence that contains each fixed-length word on a finite alphabet. In the case of length 2 strings on a binary alphabet, for example, we have 1100.

Introduction

A Universal Cycle, or Ucycle, is a non-repetitive cyclical representation of all values of a combinatorial object; a DeBruijn Sequence is the most basic example, which is a sequence that contains each fixed-length word on a finite alphabet. In the case of length 2 strings on a binary alphabet, for example, we have 1100. A Graph Universal Cycle, or Gucycle, is a particular type of Universal Cycle in which the objects in question are a family of graphs; there are objects that can be Gucycled that cannot be Ucycled.

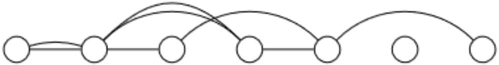
Introduction

A Universal Cycle, or Ucycle, is a non-repetitive cyclical representation of all values of a combinatorial object; a DeBruijn Sequence is the most basic example, which is a sequence that contains each fixed-length word on a finite alphabet. In the case of length 2 strings on a binary alphabet, for example, we have 1100. A Graph Universal Cycle, or Gucycle, is a particular type of Universal Cycle in which the objects in question are a family of graphs; there are objects that can be Gucycled that cannot be Ucycled. For an example of a simple Gucycle representation of an object that can be Ucycled, consider this section of a Ucycle of length 3 ternary strings, represented numerically as 0210100.



Introduction

A Universal Cycle, or Ucycle, is a non-repetitive cyclical representation of all values of a combinatorial object; a DeBruijn Sequence is the most basic example, which is a sequence that contains each fixed-length word on a finite alphabet. In the case of length 2 strings on a binary alphabet, for example, we have 1100. A Graph Universal Cycle, or Gucycle, is a particular type of Universal Cycle in which the objects in question are a family of graphs; there are objects that can be Gucycled that cannot be Ucycled. For an example of a simple Gucycle representation of an object that can be Ucycled, consider this section of a Ucycle of length 3 ternary strings, represented numerically as 0210100.



In this particular Gucycle, we consider windows of size 3, and read the number of edges between the third and first vertex, the number of edges between the first and second vertex, and the number of edges between the second and third vertex, to be the three numerical values of a particular string being represented.

Background

Definition

An Integer Composition of n is a sequence of natural numbers whose sum is n.
An Integer Partition of n is a multiset of natural numbers whose sum is n.

Background

Definition

An Integer Composition of n is a sequence of natural numbers whose sum is n.
An Integer Partition of n is a multiset of natural numbers whose sum is n.

To quote Glenn Hurlbert, "Broadly, universal cycles are special [non-repetitive] listings of combinatorial objects in which *codes for the objects* are written in an overlapping, cyclic manner."

Background

Definition

An Integer Composition of n is a sequence of natural numbers whose sum is n.
An Integer Partition of n is a multiset of natural numbers whose sum is n.

To quote Glenn Hurlbert, "Broadly, universal cycles are special [non-repetitive] listings of combinatorial objects in which *codes for the objects* are written in an overlapping, cyclic manner." Chung, Diaconis and Graham introduced Ucycles in 1990 as a generalization of DeBruijn Cycles for combinatorial objects. They showed that a Ucycle exists for an object if the overlapping of elements can be represented in a directed graph that contains an Eulerian Circuit (more on that later).

Background

Definition

An Integer Composition of n is a sequence of natural numbers whose sum is n.
An Integer Partition of n is a multiset of natural numbers whose sum is n.

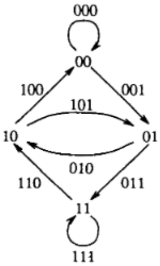
To quote Glenn Hurlbert, "Broadly, universal cycles are special [non-repetitive] listings of combinatorial objects in which *codes for the objects* are written in an overlapping, cyclic manner." Chung, Diaconis and Graham introduced Ucycles in 1990 as a generalization of DeBruijn Cycles for combinatorial objects. They showed that a Ucycle exists for an object if the overlapping of elements can be represented in a directed graph that contains an Eulerian Circuit (more on that later). Brockman, Kay, and Snively introduced Gucycles in 2009, which allowed Ucycles for objects that were represented graphically, rather than numerically.

Transition Digraph

To generate Ucycles and Gucycles, we use what's called the transition digraph, or arc digraph.

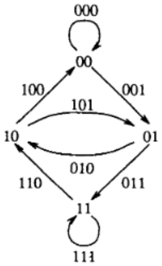
Transition Digraph

To generate Ucycles and Gucycles, we use what's called the transition digraph, or arc digraph.



Transition Digraph

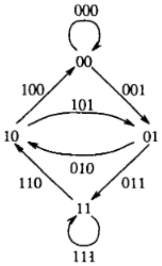
To generate Ucycles and Gucycles, we use what's called the transition digraph, or arc digraph.



The transition digraph of this DeBruijn Sequence is generated as follows: let the vertices be strings of size $n-1$. If the last $n-2$ elements of a string match the first $n-2$ elements of another string, there is a directed edge from the first vertex to the second.

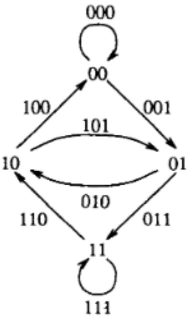
Transition Digraph

To generate Ucycles and Gucycles, we use what's called the transition digraph, or arc digraph.



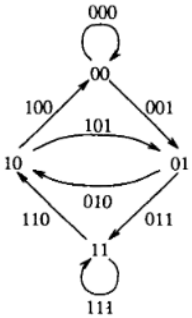
The transition digraph of this DeBruijn Sequence is generated as follows: let the vertices be strings of size $n-1$. If the last $n-2$ elements of a string match the first $n-2$ elements of another string, there is a directed edge from the first vertex to the second. From this we get our edge labels, which are the intersection of the two vertices. As such, they are length n.

Transition Digraph



This is why an Eulerian Circuit in the Transition Digraph yields a Universal Cycle.

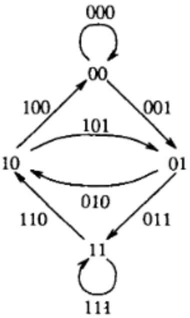
Transition Digraph



This is why an Eulerian Circuit in the Transition Digraph yields a Universal Cycle. For this graph we have:

11101000

Transition Digraph



This is why an Eulerian Circuit in the Transition Digraph yields a Universal Cycle. For this graph we have:

$$11101000, 00010111$$

down to symmetries.

Outline

This talk will explore the following:

- Gucycles of Integer Compositions

Outline

This talk will explore the following:

- Gucycles of Integer Compositions
- Ucycles of Integer Compositions using DeBruijn Sequences

Outline

This talk will explore the following:

- Gucycles of Integer Compositions
- Ucycles of Integer Compositions using DeBruijn Sequences
- Layered Metacompositions, an entirely new combinatorial object

Outline

This talk will explore the following:

- Gucycles of Integer Compositions
- Ucycles of Integer Compositions using DeBruijn Sequences
- Layered Metacompositions, an entirely new combinatorial object
- Gucycles of Integer Partitions

Graph Representation

Graphically, we choose to represent a Composition of n as a line of n vertices, with each vertex representing a 1, and an edge between two neighboring vertices representing addition.

Graph Representation

Graphically, we choose to represent a Composition of n as a line of n vertices, with each vertex representing a 1, and an edge between two neighboring vertices representing addition. For example, $(1, 3, 1, 2)$ is represented by $(1, 1+1+1, 1, 1+1)$ for $n = 7$.

Graph Representation

Graphically, we choose to represent a Composition of n as a line of n vertices, with each vertex representing a 1, and an edge between two neighboring vertices representing addition. For example, $(1, 3, 1, 2)$ is represented by $(1, 1+1+1, 1, 1+1)$ for $n = 7$.

Shifting the window to the right, we have that $(1, 3, 1, 2)$ can connect to $(3, 1, 2, 1)$, or $(3, 1, 3)$ in the transition digraph. Shifting to the left, we have that $(1, 3, 1, 2)$ can come from $(1, 1, 3, 1, 1)$, or $(2, 3, 1, 1)$.

Note that the edge labels are compositions of 8 ($(2, 3, 1, 2)$ and $(1, 1, 3, 1, 2)$ from the left, $(1, 3, 1, 3)$ and $(1, 3, 1, 2, 1)$ to the right), so any Eulerian Circuits will be Gucycles of Compositions of $(n + 1)$.

Note that the edge labels are compositions of 8 ((2, 3, 1, 2) and (1, 1, 3, 1, 2) from the left, (1, 3, 1, 3) and (1, 3, 1, 2, 1) to the right), so any Eulerian Circuits will be Gucycles of Compositions of $(n+1)$.

Theorem

Let G_C be the transition digraph of integer compositions of any n. G_C is balanced, with each vertex having an in-degree and out-degree of 2. Additionally, G_C is strongly connected. Thus, G_C is Eulerian, meaning Gucycles of Integer Compositions exist for any n.

Numerical Representation

Now, let's consider a way to encode this numerically. Before, we proved that for all $n \in \mathbb{N}$, there exist Gucycles of the compositions of n.

Numerical Representation

Now, let's consider a way to encode this numerically. Before, we proved that for all $n \in \mathbb{N}$, there exist Gucycles of the compositions of n. However, if we can encode these graphs numerically, we can prove that Ucycles of the compositions of n exist.

Numerical Representation

Now, let's consider a way to encode this numerically. Before, we proved that for all $n \in \mathbb{N}$, there exist Gucycles of the compositions of n. However, if we can encode these graphs numerically, we can prove that Ucycles of the compositions of n exist. Let 1 represent two neighboring vertices with an edge between them, and let 0 represent two neighboring vertices without an edge.

Numerical Representation

Now, let's consider a way to encode this numerically. Before, we proved that for all $n \in \mathbb{N}$, there exist Gucycles of the compositions of n. However, if we can encode these graphs numerically, we can prove that Ucycles of the compositions of n exist. Let 1 represent two neighboring vertices with an edge between them, and let 0 represent two neighboring vertices without an edge. This construction allows you to represent a Composition Graph on n vertices as a binary string of length $(n-1)$.

Numerical Representation

Now, let's consider a way to encode this numerically. Before, we proved that for all $n \in \mathbb{N}$, there exist Gucycles of the compositions of n. However, if we can encode these graphs numerically, we can prove that Ucycles of the compositions of n exist. Let 1 represent two neighboring vertices with an edge between them, and let 0 represent two neighboring vertices without an edge. This construction allows you to represent a Composition Graph on n vertices as a binary string of length $(n-1)$. Let's see how this looks for $n = 4$. Recall that 11101000 is a DeBruijn Sequence of length 3 binary strings.

$$11101000 \implies (4) \to (3,1) \to (2,2) \to (1,2,1) \to (2,1,1) \to (1,1,1,1) \to (1,1,2) \to (1,3) \to (4)$$

Numerical Representation

Now, let's consider a way to encode this numerically. Before, we proved that for all $n \in \mathbb{N}$, there exist Gucycles of the compositions of n. However, if we can encode these graphs numerically, we can prove that Ucycles of the compositions of n exist. Let 1 represent two neighboring vertices with an edge between them, and let 0 represent two neighboring vertices without an edge. This construction allows you to represent a Composition Graph on n vertices as a binary string of length $(n-1)$. Let's see how this looks for $n = 4$. Recall that 11101000 is a DeBruijn Sequence of length 3 binary strings.

$$11101000 \implies (4) \to (3,1) \to (2,2) \to (1,2,1) \to (2,1,1) \to \\ (1,1,1,1) \to (1,1,2) \to (1,3) \to (4)$$

Therefore, since we already know DeBruijn Sequences exist for binary strings of any length, Ucycles must exist for compositions of any $n \in \mathbb{N}$.

Layered Metacompositions

For now, let's forget the graph representation and define our strings as follows: 1 represents addition, and 0 represents separation; this is the same definition as before, just in different terms.

Layered Metacompositions

For now, let's forget the graph representation and define our strings as follows: 1 represents addition, and 0 represents separation; this is the same definition as before, just in different terms. Now, let's consider what expanding our alphabet could look like. Let's have 2 represent a stronger level of separation, where 0 represents a weaker level of separation.

Layered Metacompositions

For now, let's forget the graph representation and define our strings as follows: 1 represents addition, and 0 represents separation; this is the same definition as before, just in different terms. Now, let's consider what expanding our alphabet could look like. Let's have 2 represent a stronger level of separation, where 0 represents a weaker level of separation. Now imagine breaking the composition down into layers, where at each level we look at a different level of separation.

Layered Metacompositions

For now, let's forget the graph representation and define our strings as follows: 1 represents addition, and 0 represents separation; this is the same definition as before, just in different terms. Now, let's consider what expanding our alphabet could look like. Let's have 2 represent a stronger level of separation, where 0 represents a weaker level of separation. Now imagine breaking the composition down into layers, where at each level we look at a different level of separation. If we look at a string from the first layer, 0 and 2 both function by separating the composition into parts. But if we look at the string from the second layer, then we'll ignore the weakest level of separation.

Layered Metacompositions

For now, let's forget the graph representation and define our strings as follows: 1 represents addition, and 0 represents separation; this is the same definition as before, just in different terms. Now, let's consider what expanding our alphabet could look like. Let's have 2 represent a stronger level of separation, where 0 represents a weaker level of separation. Now imagine breaking the composition down into layers, where at each level we look at a different level of separation. If we look at a string from the first layer, 0 and 2 both function by separating the composition into parts. But if we look at the string from the second layer, then we'll ignore the weakest level of separation. This is how we'll define Layered Metacompositions. For $j \geq 2$, j represents the $(j-1)$'th level of separation; 0 represents the weakest level of separation. Looking at the jth layer of a composition means ignoring the $(j-1)$ weakest levels of separation, and treating all remaining levels the same.

Layered Metacompositions

For now, let's forget the graph representation and define our strings as follows: 1 represents addition, and 0 represents separation; this is the same definition as before, just in different terms. Now, let's consider what expanding our alphabet could look like. Let's have 2 represent a stronger level of separation, where 0 represents a weaker level of separation. Now imagine breaking the composition down into layers, where at each level we look at a different level of separation. If we look at a string from the first layer, 0 and 2 both function by separating the composition into parts. But if we look at the string from the second layer, then we'll ignore the weakest level of separation. This is how we'll define Layered Metacompositions. For $j \geq 2$, j represents the $(j-1)$'th level of separation; 0 represents the weakest level of separation. Looking at the jth layer of a composition means ignoring the $(j-1)$ weakest levels of separation, and treating all remaining levels the same. For example, $021 \implies (1, 1; 1+1)$. On the first level, $021 \implies (1, 1, 2)$. On the second level, $021 \implies (2, 2)$.

Layered Metacompositions

For now, let's forget the graph representation and define our strings as follows: 1 represents addition, and 0 represents separation; this is the same definition as before, just in different terms. Now, let's consider what expanding our alphabet could look like. Let's have 2 represent a stronger level of separation, where 0 represents a weaker level of separation. Now imagine breaking the composition down into layers, where at each level we look at a different level of separation. If we look at a string from the first layer, 0 and 2 both function by separating the composition into parts. But if we look at the string from the second layer, then we'll ignore the weakest level of separation. This is how we'll define Layered Metacompositions. For $j \geq 2$, j represents the $(j-1)$'th level of separation; 0 represents the weakest level of separation. Looking at the jth layer of a composition means ignoring the $(j-1)$ weakest levels of separation, and treating all remaining levels the same. For example, $021 \implies (1,1;1+1)$. On the first level, $021 \implies (1,1,2)$. On the second level, $021 \implies (2,2)$. Thus, in the same way as before, we can represent these Compositions-in-Compositions of n as DeBruijn Sequences with alphabet size j and window size $(n-1)$.

Integer Partitions

Now, we'll look at Gucycles of Integer Partitions.

Integer Partitions

Now, we'll look at Gucycles of Integer Partitions. Recall that an integer partition is a multiset of natural numbers whose sum is n; this definition is very close to that of an integer composition of n, with the key difference being that a multiset is unordered. This means the process of finding Gucycles for these will be slightly more complicated than with other objects.

Integer Partitions

Now, we'll look at Gucycles of Integer Partitions. Recall that an integer partition is a multiset of natural numbers whose sum is n; this definition is very close to that of an integer composition of n, with the key difference being that a multiset is unordered. This means the process of finding Gucycles for these will be slightly more complicated than with other objects. First, we'll start by defining our family of graphs. An Integer Partition graph of n is the union of t complete graphs on s vertices such that $\sum_{\gamma=1}^{t} s_\gamma = n$.

Integer Partitions

Now, we'll look at Gucycles of Integer Partitions. Recall that an integer partition is a multiset of natural numbers whose sum is n; this definition is very close to that of an integer composition of n, with the key difference being that a multiset is unordered. This means the process of finding Gucycles for these will be slightly more complicated than with other objects. First, we'll start by defining our family of graphs. An Integer Partition graph of n is the union of t complete graphs on s vertices such that $\sum_{\gamma=1}^{t} s_\gamma = n$. Essentially, this just defines any part of n as a complete graph on that many vertices. We use three non-unique characteristics to describe a partition graph: n, the natural number we're separating into partitions,

Integer Partitions

Now, we'll look at Gucycles of Integer Partitions. Recall that an integer partition is a multiset of natural numbers whose sum is n; this definition is very close to that of an integer composition of n, with the key difference being that a multiset is unordered. This means the process of finding Gucycles for these will be slightly more complicated than with other objects. First, we'll start by defining our family of graphs. An Integer Partition graph of n is the union of t complete graphs on s vertices such that $\sum_{\gamma=1}^{t} s_\gamma = n$. Essentially, this just defines any part of n as a complete graph on that many vertices. We use three non-unique characteristics to describe a partition graph: n, the natural number we're separating into partitions, t, the total number of parts of the graph,

Integer Partitions

Now, we'll look at Gucycles of Integer Partitions. Recall that an integer partition is a multiset of natural numbers whose sum is n; this definition is very close to that of an integer composition of n, with the key difference being that a multiset is unordered. This means the process of finding Gucycles for these will be slightly more complicated than with other objects. First, we'll start by defining our family of graphs. An Integer Partition graph of n is the union of t complete graphs on s vertices such that $\sum_{\gamma=1}^{t} s_\gamma = n$. Essentially, this just defines any part of n as a complete graph on that many vertices. We use three non-unique characteristics to describe a partition graph: n, the natural number we're separating into partitions, t, the total number of parts of the graph, and d, the number of distinct parts of the graph, which is equal to the number of values s takes on.

Integer Partitions

Now, we'll look at Gucycles of Integer Partitions. Recall that an integer partition is a multiset of natural numbers whose sum is n; this definition is very close to that of an integer composition of n, with the key difference being that a multiset is unordered. This means the process of finding Gucycles for these will be slightly more complicated than with other objects. First, we'll start by defining our family of graphs. An Integer Partition graph of n is the union of t complete graphs on s vertices such that $\sum_{\gamma=1}^{t} s_\gamma = n$. Essentially, this just defines any part of n as a complete graph on that many vertices. We use three non-unique characteristics to describe a partition graph: n, the natural number we're separating into partitions, t, the total number of parts of the graph, and d, the number of distinct parts of the graph, which is equal to the number of values s takes on. We denote this as $p(n, t, d)$.

Connectedness

We say that p, a partition graph of n, is connected to p', another partition of n in the transition digraph, if p transforms into p' by the following process:

Connectedness

We say that p, a partition graph of n, is connected to p', another partition of n in the transition digraph, if p transforms into p' by the following process:

- Add a vertex to p. This transforms p into a partition of $(n+1)$.

Connectedness

We say that p, a partition graph of n, is connected to p', another partition of n in the transition digraph, if p transforms into p' by the following process:

- Add a vertex to p. This transforms p into a partition of $(n+1)$.
- Choose whether to connect the new vertex to any of the existing parts of p, or let it remain a lone vertex.

Connectedness

We say that p, a partition graph of n, is connected to p', another partition of n in the transition digraph, if p transforms into p' by the following process:

- Add a vertex to p. This transforms p into a partition of $(n+1)$.
- Choose whether to connect the new vertex to any of the existing parts of p, or let it remain a lone vertex.
- Delete any vertex of p.

Connectedness

We say that p, a partition graph of n, is connected to p', another partition of n in the transition digraph, if p transforms into p' by the following process:

- Add a vertex to p. This transforms p into a partition of $(n+1)$.
- Choose whether to connect the new vertex to any of the existing parts of p, or let it remain a lone vertex.
- Delete any vertex of p.

From this definition, two things are evident:

Theorem

Let G_P be the transition digraph of integer partitions of any n.

- G_P is strongly connected.
- Each edge is bidirectional, so G_P is balanced.

Connectedness

We say that p, a partition graph of n, is connected to p', another partition of n in the transition digraph, if p transforms into p' by the following process:

- Add a vertex to p. This transforms p into a partition of $(n+1)$.
- Choose whether to connect the new vertex to any of the existing parts of p, or let it remain a lone vertex.
- Delete any vertex of p.

From this definition, two things are evident:

Theorem

Let G_P be the transition digraph of integer partitions of any n.

- G_P is strongly connected.
- Each edge is bidirectional, so G_P is balanced.

Thus, G_P is Eulerian, meaning Gucycles of Integer Partitions exist for any n.

In and Out Degree

Let's find the in and out degree of each vertex by going back over how these edges are drawn, and counting how many distinct ways there are to do so.

In and Out Degree

Let's find the in and out degree of each vertex by going back over how these edges are drawn, and counting how many distinct ways there are to do so.

- *Add a vertex to p.* Since integer partitions are unordered, it doesn't matter where this goes. Thus, there is one way to do this.

In and Out Degree

Let's find the in and out degree of each vertex by going back over how these edges are drawn, and counting how many distinct ways there are to do so.

- *Add a vertex to p.* Since integer partitions are unordered, it doesn't matter where this goes. Thus, there is one way to do this.
- *Choose whether to connect the new vertex to any of the existing parts of p, or let it remain a lone vertex.* Recall that d is the number of distinct parts of p. Since there is one way to leave a lone vertex, and d distinct ways to connect a vertex to one of the parts of p, there are $(d + 1)$ ways to do this.

In and Out Degree

Let's find the in and out degree of each vertex by going back over how these edges are drawn, and counting how many distinct ways there are to do so.

- *Add a vertex to p.* Since integer partitions are unordered, it doesn't matter where this goes. Thus, there is one way to do this.
- *Choose whether to connect the new vertex to any of the existing parts of p, or let it remain a lone vertex.* Recall that d is the number of distinct parts of p. Since there is one way to leave a lone vertex, and d distinct ways to connect a vertex to one of the parts of p, there are $(d+1)$ ways to do this.
- *Delete any vertex.* For any $p(n, t, d)$, there are d distinct ways to delete a vertex from any part. However, after Step 2 we have multiple different partitions of $(n+1)$ that could potentially have different values of d. Thus, we don't have a closed form for this.

In and Out Degree

The best we can do is this:

Theorem

Let p be an integer partition graph with $d(p)$ distinct parts, and let α be the set of partitions of $(n+1)$ that are created after Step 2. We have then that the number of partitions that p is connected to is equal to

$$\sum_{s=1}^{d(p)+1} d(\alpha_s)$$

In and Out Degree

The best we can do is this:

Theorem

Let p be an integer partition graph with $d(p)$ distinct parts, and let α be the set of partitions of $(n+1)$ that are created after Step 2. We have then that the number of partitions that p is connected to is equal to

$$\sum_{s=1}^{d(p)+1} d(\alpha_s)$$

By our definition of the edges, we have that each vertex has at least one loop to itself.

In and Out Degree

The best we can do is this:

Theorem

Let p be an integer partition graph with $d(p)$ distinct parts, and let α be the set of partitions of $(n+1)$ that are created after Step 2. We have then that the number of partitions that p is connected to is equal to

$$\sum_{s=1}^{d(p)+1} d(\alpha_s)$$

By our definition of the edges, we have that each vertex has at least one loop to itself. However, since most of these loops would have the same labels as other loops, and loops don't really affect Eulerian Circuits, we choose to ignore all loops that don't have a unique edge label.

Thank you!

Funding for this presentation was granted by the Office of Undergraduate Research and Creative Activities

References

Brockman, G., Kay, B., & Snively, E. E. (2010). On universal cycles of labeled graphs. The Electronic Journal of Combinatorics, 17(1). https://doi.org/10.37236/276 Cantwell, A., Geraci, J., Godbole, A., & Padilla, C. (2021). Graph universal cycles of combinatorial objects. Advances in Applied Mathematics, 127, 102166. https://doi.org/10.1016/j.aam.2021.102166 Chung, F., Diaconis, P., & Graham, R. (1992). Universal cycles for combinatorial structures. Discrete Mathematics, 110(1–3), 43–59. https://doi.org/10.1016/0012-365x(92)90699-g