**Maximum score:** 120 points.
**Instructions:** For this test, you work in teams to solve a multi-part, proof-oriented question. Problems that use the words "compute," "list," or "draw" require only an answer; no explanation or proof is needed. Unless otherwise stated, all other questions require explanation or proof. The problems are ordered by content, *not difficulty*. The difficulties of the problems are generally indicated by the point values assigned to them; it is to your advantage to attempt problems throughout the test. In your solution for a given problem, you may cite the statements of earlier problems (but not later ones) without additional justification, even if you haven't solved them.
**No Calculators.**

# 1 Combinatorial Classics: Trees, Permutations, Partitions

## 1.1 Graphs and Trees

A graph provides a pictoral way to represent relationships between objects. There is a lot of combinatorics that can be done on graphs, so we introduce them here in this section.
A graph consists of some vertices, which are points in the plane, and edges in between them. Some examples of graphs are in Figure 1. A compact way to *describe* a graph is using set notation. Particularly, a graph

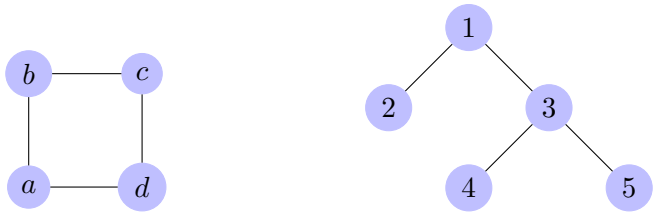


Figure 1: Graphs $G_1$ (left) and $G_2$ (right)

$G$ comprises of a set of vertices $V$, described by the labels on the vertices, and a set of edges $E$, described by the vertices being connected by the edge.

**Example 1.1.** For the graph $G_1$ in Figure 1, we say that $G_1 = (V_1, E_1)$ is a graph. Here $V_1 = \{a, b, c, d\}$ is the vertex set. Additionally, $E_1$ is the edge set that describes connections between two vertices. We represent these connections as $\{\cdot, \cdot\}$. In the case of $G_1$, this is $E_1 = \Big\{\{a, b\}, \{b, c\}, \{c, d\}, \{d, a\}\Big\}$.

**Example 1.2.** Now consider the graph $G_2 = (V_2, E_2)$. The vertex set is $V_2 = \{1, 2, 3, 4, 5\}$. The edge set is $\Big\{\{1, 2\}, \{1, 3\}, \{3, 4\}, \{3, 5\}\Big\}$.

**Question 1.1.**

(a) (**2**) Let $V = \{a, b, c, d, e\}$ and $E = \Big\{\{a, b\}, \{b, c\}, \{c, d\}, \{d, e\}, \{e, a\}, \{a, d\}\Big\}$. Draw the corresponding graph $G = (V, E)$ (with labels on vertices).

(b) (**2**) A complete graph on $n$ vertices, denoted by $K_n = (V_n, E_n)$, is a graph where any two (distinct) vertices are connected by exactly one edge. Consider $K_5 = (V_5, E_5)$, or the complete graph with 5 vertices. Express $V_5$ and $E_5$ as sets; no justification is needed. You may choose any set of 5 labels for the vertex set.

(c) (**3**) Draw the graph $K_5$ from part (b). Compute the number of edges $K_5$ has.

(d) (**2**) More generally, how many edges will the complete graph $K_n$ have?

This suggests a more general definition for graphs:

**Definition 1.1.1.** A **graph** is a (ordered) pair $G = (V, E)$ where

- $V = \{v_1, v_2, \ldots, v_n\}$ is a set of **vertices**.
- $E$ is a set of **edges** in between vertices, represented as

$$E = \Big\{\{v_j, v_k\} : v_j, v_k \in V \text{ are joined by an edge}\Big\}$$

While this formal definition is very useful in its own world, we will mostly be looking at the picture form of graphs: drawing vertices as points in the plane and edges as line connections between these points. An important sub-family of graphs is trees.

**Definition 1.1.2.** A **tree** is a graph where any two vertices are connected by exactly one path. In other words, a tree is a graph without any loops or cycles.

Trees are usually drawn in the up-to-down fashion of graph $G_2$ in Figure 1 and the graphs in Figure 2. We describe some specific kinds of trees.

**Definition 1.1.3.** A **rooted tree** is a tree in which one specific vertex has been chosen as the root, with other vertices below it.

**Definition 1.1.4.** A **child** of a vertex $v$ is a vertex connected to and below $v$.

**Definition 1.1.5.** A **$n$-ary tree** is a rooted tree where each vertex has 0 or $n$ children. If a vertex has 0 children, it is called a leaf. If a vertex has $n$ children, it is called a node. We will specifically focus on 2-nary trees, also called **binary trees**.

**Example 1.3.** For example, the children of $c$ in the blue rooted tree in Figure 2 are $d$ and $e$. The vertex **A** is the root. Since each vertex has 0 or 2 children, the tree is actually a 2-nary or *binary* tree.

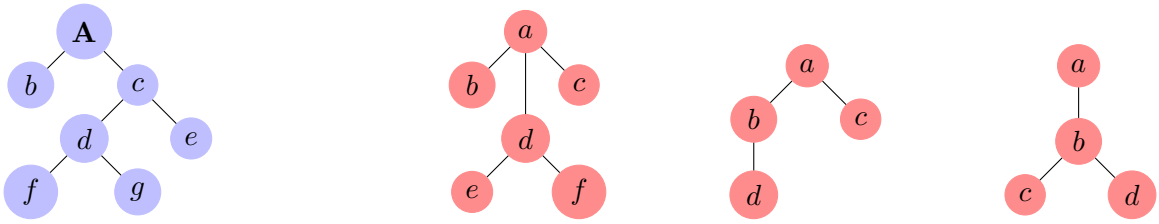


Figure 2: A rooted tree (left, blue) and some other trees (right, red)

**Example 1.4.** Here are all of the binary trees with 2 nodes.

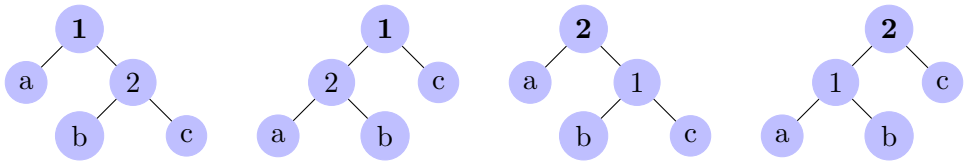


Figure 3: Binary Trees with 2 nodes

**Question 1.2 (2).** Compute the number of binary (2-nary) rooted trees with 3 nodes.

**Question 1.3 (3).** If a tree has $n$ vertices, how many edges does it have? Prove your result. (*Hint:* Use induction.)

## 1.2 Permutations

The next famous object from combinatorics that we will talk about is permutations. Roughly, a permutation rearranges the elements of the set in some way. For example, $\sigma_1$ acts on (rearranges the elements of) $\{1, 2, 3\}$ by sending $1 \mapsto 2, 2 \mapsto 3$, and $3 \mapsto 1$ is a permutation. Based on this, we can provide a more general definition.

**Definition 1.2.1.** A **permutation** $\sigma$ of a (finite) set $S$ is a bijective function (one-to-one correspondence) from $S$ to itself that rearranges the elements. In other words, $\sigma$ is a function that sends every element of $S$ to some element of $S$ (possibly the same element), and no two elements are sent to the same element.

Typically, we think of permutations as acting on the set $S = \{1, 2, \cdots, n\}$, denoted $[n]$, of the first $n$ natural numbers.

**Example 1.5.** The following are some examples (and a non-example) of permutations:

- $\sigma_2 : [5] \to [5]$ given by $\sigma_2(1) = 1$, $\sigma_2(2) = 5$, $\sigma_2(4) = 3$, $\sigma_2(3) = 2$, and $\sigma_2(5) = 4$ is a permutation.
- $\sigma_3 : [4] \to [4]$ given by $\sigma_3(i) = i$ for $1 \leq i \leq 4$ is also a permutation. This permutation has a special name; see the next definition.
- However, $\sigma_4 : [3] \to [3]$ given by $\sigma_4(3) = 1$, $\sigma_4(2) = 1$, $\sigma_4(1) = 2$ is **NOT** a permutation. This is because 3 and 2 both map to 1 under $\sigma_4$. The elements are **not** *rearranged*, but rather 3 and 2 are superimposed by $\sigma_4$.

**Definition 1.2.2.** For any natural number $n$, the permutation $e_n : [n] \to [n]$ given by $e_n(i) = i$ for $1 \leq i \leq n$ is called the **identity** permutation. We use $e$ to denote $e_n$ for any $n$.

**Definition 1.2.3.** We usually express permutations in the **one-line notation**

$$\sigma = \Big(\sigma(1) \quad \sigma(2) \quad \sigma(3) \quad \cdots \quad \sigma(n)\Big)'.$$

The prime ($'$) at the end of the parenthesis is used to distinguish between another kind of notation that is typically used for permutations.

**Example 1.6.** From Example 1.5, the permutation $\sigma_2$ can be written in one-line notation as $\sigma_2 = (15324)'$.

**Definition 1.2.4.** The **composition** of permutations $\sigma \circ \tau$, for permutations $\sigma$ and $\tau$ of $[n]$, is given by:

$$\sigma \circ \tau = \Big(\sigma(\tau(1)) \quad \sigma(\tau(2)) \quad \cdots \quad \sigma(\tau(n))\Big)'.$$

For example, consider $\sigma = (3214)'$ and $\tau = (2314)'$. Then $\sigma \circ \tau = (\sigma(2)\,\sigma(3)\,\sigma(1)\,\sigma(4))' = (2134)'$.

**Question 1.4.** For $\sigma = (12345)'$ and $\tau = (51324)'$, compute the following:

(a) **(1)** $\sigma \circ \tau$ (b) **(1)** $\tau \circ \sigma$ (c) **(2)** $\sigma^3 = \sigma \circ \sigma \circ \sigma$.

**Question 1.5.** An element $j \in [n]$ is fixed under $\sigma$ if $\sigma(j) = j$.

(a) (**1.5**) Compute the number of permutations $\sigma$ of [4] that fix no elements of [4].

(b) (**1.5**) How many permutations $\tau$ of [5] fix exactly one element of [5]?

(c) (**3**) How many permutations of $[n]$ are such that all but 2 elements of $[n]$ remain fixed?

**Question 1.6** (**4**)**.** A permutation $\sigma = \begin{pmatrix}\sigma(1) & \sigma(2) & \cdots & \sigma(n)\end{pmatrix}'$ is said to be **unimodal** if the numbers in its one-line notation increase at first then decrease with only one *peak*.
For instance,

$$\begin{pmatrix}1 & 2 & 4 & \mathbf{8} & 7 & 6 & 5 & 3\end{pmatrix}' \text{ is unimodal.}$$

More specifically, a permutation is unimodal if there exists $k$, with $2 \leq k \leq n-1$, such that $\sigma(1) < \sigma(2) < \cdots < \sigma(k)$ and $\sigma(k) > \sigma(k+1) > \cdots > \sigma(n)$. For $n \geq 3$, how many permutations of $[n]$ are unimodal?

**Question 1.7** (**3**)**.** Compute the number of permutations of [4] that are **involutions**; i.e, compute the number of permutations $\sigma$ of [4] such that $\sigma \circ \sigma = e$, where $e$ is the identity.

**Question 1.8** (**5**)**.** Consider a permutation $\sigma = (24315)'$ of [5]. Notice that $1 \mapsto 2 \mapsto 4 \mapsto 1$ and $3 \mapsto 5 \mapsto 3$ (That is, $\sigma(1) = 2, \sigma(2) = 4, \sigma(4) = 1$ and $\sigma(3) = 5, \sigma(5) = 3$). Thus our permutation has two *cycles*. It can then be written in **cycle notation** $\sigma = (124)(35)$. Show that every permutation of $[n]$ can be written in cycle notation. (*Hint:* A pigeonhole argument works well here.)

The advantage of cycle notation is that it makes the effect of repeated composition $\sigma \circ \sigma \circ \sigma \cdots \circ \sigma$ much more apparent. For instance, if $\sigma = (124)(3)(5)$, then $\sigma^2 = \sigma \circ \sigma = (142)(3)(5)$ and $\sigma^3 = (1)(2)(4)(3)(5) =$ identity. Note that we are **NOT** adding a prime at the end of the parentheses for cyclic notation.

### 1.3 Partitions of integers and sets

Partitions are an absolute number theory classic, and have been famous for spawning a wide and diverse variety of math. We can partition integers and sets, both of which carry a lot of cool math with them. We begin with the former.

**Definition 1.3.1.** An **integer partition** (or simply partition) of $n$ is a way of decomposing $n$ into positive integer summands, where the order of the summands does not matter.

For instance,

$$10 = 3 + 3 + 2 + 1 + 1$$

is a partition of 10 (order does not matter, so we typically go with descending order). Here are the partitions of 4:

$$4 = 4 = 3 + 1 = 2 + 2 = 2 + 1 + 1 = 1 + 1 + 1 + 1$$

**Definition 1.3.2.** The **partition function** $p(n)$ is the number of partitions $n$ has. As evident from above, we have $p(4) = 5$.

**Question 1.9** (**2**)**.** Compute $p(7)$.

**Question 1.10.**

(a) (**1**) Let $p_e(n)$ be the number of ways to partition $n$ into summands that are all even. Compute $p_e(7)$.

(b) (**1.5**) Similarly, compute $p_o(7)$, or the number of ways to partition 7 into summands that are odd (for instance, $7 = 5 + 1 + 1$).

(c) (**1.5**) Finally, compute $p_d(7)$, which is the number of ways to partition 7 into distinct summands (for example, $7 = 4 + 3$ works, but not $7 = 5 + 1 + 1$).

**Question 1.11** (**5**)**.** Prove the following remarkable fact: The number of partitions of $n$ into distinct summands is equal to the number of partitions into odd summands; that is, $p_d(n) = p_o(n)$. (*Hint:* Use the fact that every integer can be uniquely represented as a power of two times an odd integer.)

**Question 1.12** (**4**)**.** Let $p(n,k)$ denote the number of partitions of $n$ that have exactly $k$ parts. Show that

$$p(n,k) = p(n-1,k-1) + p(n-k,k).$$

Next, we talk about partitions of sets.

**Definition 1.3.3.** A **set partition** of $X_n = \{1,2,\ldots,n\}$ is a grouping of its elements into disjoint and non-empty subsets $S_i$ for $1 \le i \le k$ such that $S_1 \cup S_2 \cup \cdots \cup S_k = X_n$. Note that the order of the sets and the order of the elements within a set do not matter. We denote by $\mathrm{Par}(n)$ the number of set partitions of $X_n$.

**Example 1.7.** For $n = 3$, we have

$$\{1,2,3\} = \{1\} \cup \{2\} \cup \{3\} = \{1,2\} \cup \{3\} = \{1,3\} \cup \{2\} = \{2,3\} \cup \{1\} = \{1,2,3\}$$

as all the possible set partitions, so $\mathrm{Par}(3) = 5$.

**Question 1.13** (**2**)**.** Compute $\mathrm{Par}(4)$.

## 2 Generating Functions

Often in combinatorics and other fields, we find ourselves working with sequences $a_0, a_1, a_2, \ldots$, denoted $(a_n)_{n\ge 0}$. *Generating functions* furnish us with a powerful tool for working with sequences, and in many cases, discovering new properties of them. There are two types of generating functions are often found in combinatorics: *ordinary* generating functions and *exponential* generating functions. When it comes to labelled structures (which is our focus for this power round), the latter is very useful.

**Definition 2.0.1.** Let $(a_n)_{n\ge 0}$ be a sequence of real numbers. The **exponential generating function (EGF)** of $(a_n)$ is given by

$$A(z) = \sum_{n=0}^{\infty} a_n \frac{z^n}{n!}.$$

**Note:** It is important to note that generating functions are *formal* power series. (If you have not taken Calculus, you can think of a power series as an infinite polynomial.) This means that we pay no mind to whether or not these sums converge.

**Example 2.1.** A simple sequence we can consider is $1, 1, \ldots$, given by $e_n = 1$ for all $n$. In this case, we have

$$E(z) = \sum_{n=0}^{\infty} 1 \cdot \frac{z^n}{n!} = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \cdots.$$

This is an important function called the *exponential function*, and we write it as $\exp(z)$ or $e^z$, where $e \approx 2.718$.

**Example 2.2.** Consider the sequence $P_n =$ the number of subsets of $\{1,\ldots,n\}$. When forming a subset of $\{1,2,3\ldots,n\}$, note that each element has the choice of either being in the subset or not being in the subset. Thus, there are two choices for each element, and a total of $n$ elements. The number of subsets then is

$$P_n = \underbrace{2 \cdot 2 \cdot 2 \cdots 2}_{n \text{ times}} = 2^n.$$

The EGF $P(z)$ of $P_n$ then is

$$P(z) = \sum_{n=0}^{\infty} 2^n \cdot \frac{z^n}{n!} = \sum_{n=0}^{\infty} \frac{(2z)^n}{n!} = e^{2z}.$$

**Question 2.1.** None of your final answers should be in summation notation.

(a) (**2**) Calculate the EGF $\mathrm{Per}(z)$ of $p_n$ = the number of permutations of $\{1, \ldots, n\}$ and justify your answer. *Hint:* Use the formula for the sum of an infinite geometric series. You may ignore any conditions on the common ratio usually involved.

(b) (**3**) Let $t_n$ be the number of ways in which $n$ distinct people can be arranged into pairs, and let $T(z)$ be the corresponding EGF. Then we have

$$t_n = \begin{cases} \dfrac{(2k)!}{2^k \cdot k!} & n = 2k \text{ is even} \\ \\ 0 & n = 2k+1 \text{ is odd} \end{cases}.$$

Prove that $T(z) = e^{z^2/2}$.

(c) (**2**) Calculate the EGF of $(a_n)$, where for a given positive integer $a$, we define

$$a_n = \begin{cases} \frac{1}{(a-n)!} & \text{if } 0 \le n \le a \\ 0 & \text{else} \end{cases}.$$

Justify your answer.

Generating functions offer us a compact way of encapsulating an entire sequence $(a_n)$. Given the generating function of a sequence, we can "read off" the elements of the sequence by calculating the coefficients of the power series. In particular, if $A(z)$ is the EGF of $(a_n)$, then $a_k$ is the "coefficient of $z^k/k!$", which we denote

$$a_k = \left[\frac{z^k}{k!}\right] A(z).$$

Part of the niceness of generating functions is the fact that often times, arithmetic operations with generating functions correspond to meaningful operations on the sequences. For example, let $(a_n)$ and $(b_n)$ be sequences. Then we have

$$A(z) + B(z) = \left(\sum_{n=0}^{\infty} a_n \frac{z^n}{n!}\right) + \left(\sum_{n=0}^{\infty} b_n \frac{z^n}{n!}\right) = \sum_{n=0}^{\infty} (a_n + b_n) \frac{z^n}{n!}.$$

Thus, if $c_n = a_n + b_n$, then $C(z) = A(z) + B(z)$.
One might be tempted to assume that if $C(z) = A(z)B(z)$, then $c_n = a_n b_n$. However, this is not the case:

**Question 2.2** (**4**)**.** Let $C(z) = \sum_{n=1}^{\infty} c_n \frac{z^n}{n!}$. Show that if $C(z) = A(z)B(z)$, then

$$c_n = \sum_{m=0}^{n} \binom{n}{m} a_m b_{n-m}.$$

It is interesting to see the binomial coefficient $\binom{n}{m}$ appear in the above formula. Recalling that $\binom{n}{m}$ is the number of subsets of $\{1, \ldots, n\}$ with $m$ elements, we may take this to suggest that multiplying EGFs is somehow connected to choosing a subset of a set. (This will be made concrete later with combinatorial species.)

**Question 2.3 (3).** Use the fact that $P(z) = e^{2z}$ from Example 2.2 and use the result of Question 2.2 to prove that

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n.$$

**Question 2.4 (2).** Prove Question 2.3 using a counting argument.

**Definition 2.0.2.** In addition to adding and multiplying generating functions, we can also *compose* them. While there is an explicit formula for $c_n$ given $C(z) = A(B(z))$, we refrain from discussing it since it is computationally intensive.

# 3 Combinatorial Species

## 3.1 What is a Combinatorial Species?

A very common theme in combinatorics is that combinatorial objects (like graphs) are defined either implicitly or in terms of other objects. There are many instances of this:

- Remove the root of a binary tree and you have two binary trees!
- Permutations can be written in cycle notation (see Problem 1.8): that basically makes them a set of cycles of different sizes.

In traditional combinatorics, we often use recursion to take advantage of these relations. That being said, this approach can be messy and depends on the size $n$ of the object (for example, number of vertices). What if there was a way to get rid of $n$ here? That is, what if we create relationships between two families of structures?
Combinatorial species are a way for us to create these families, and then explore bijections between them.

**Example 3.1.** Let us start with an example. Given any set $U$, the combinational species $\mathsf{Per}$ is a rule that produces the permutations of $U$. For instance, here is what $\mathsf{Per}$ produces for $U = \{1, 2, 3\}$:

$$\{1,2,3\}$$
$$\Downarrow \mathsf{Per}[\{1,2,3\}]$$

$$e = (123)' \qquad \sigma_1 = (132)' \qquad \sigma_2 = (213)'$$

$$\sigma_3 = (231)' \qquad \sigma_4 = (312)' \qquad \sigma_5 = (321)'$$

Figure 4: Action of the species $\mathsf{Per}$ on the set $\{1, 2, 3\}$ produces permutations of $\{1, 2, 3\}$

Here, we denote by $\mathsf{Per}[U]$ the resulting set $\{e, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5\}$ of permutations of $U$. We call any element $p \in \mathsf{Per}[U]$ a $\mathsf{Per}$-structure on $U$. So $\sigma_1 = (132)'$ is a $\mathsf{Per}$ structure on $\{1, 2, 3\}$.

**Definition 3.1.1** (Combinatorial Species)**.** A **combinatorial species** $\mathcal{F}$ is a rule that produces, for each set $U$, a set $\mathcal{F}[U]$ of combinatorial objects which use $U$ as labels.
Additionally, relabelling the set $U$ does not change the structure of $\mathcal{F}[U]$. (For example, relabelling the $\{1, 2, 3\}$ with $\{4, 5, 6\}$ in Example 3.1 does not change the inherent properties of $\sigma_1$.) For our purposes, this will be more of a technical condition that will be ignored. An element $s \in \mathcal{F}[U]$ associated to the species is called an $\mathcal{F}$ **structure** on $U$.

Thus, a combinatorial species produces a family of structures $\mathcal{F}[U]$ out of sets $U$. The definition has a lot to unpack, so we provide several examples.

- The species $\mathcal{B}$ of rooted binary trees: For example, if $U = \{1, 2, 3, 4, 5\}$, the *rule* $\mathcal{B}$ produces the set $\mathcal{B}[U]$ of binary trees with 5 **nodes**, labelled 1 through 5. **Please carefully note that we are saying nodes and not vertices!** You can assign arbitrary labels to the leaves. An element $b \in \mathcal{B}[U]$ is a binary tree with 5 nodes labelled by $U$.
- The species $\mathcal{T}$ of trees. For any finite set $U$, an element $t \in \mathcal{T}[U]$ is a tree of $|U| = n$ nodes, with the labels being elements of $U$.
- The single element species or atomic species $\mathcal{Z}$ defined by

$$\mathcal{Z}[U] = \begin{cases} \{U\} & \text{if } |U| = 1 \\ \emptyset & \text{otherwise} \end{cases}.$$

  This species is used to single out one point from a combinatorial object.
- The one-species 1 represents the empty set, and is given by

$$1[U] = \begin{cases} \{U\} & \text{if } U = \emptyset \\ \emptyset & \text{otherwise} \end{cases}.$$

- $\mathsf{Set}$ is the species of sets. $\mathsf{Set}[U] = \{U\}$ for all finite sets $U$.
- $\mathsf{Cyc}$ is the species of cyclic permutations, or permutations with only one cycle in cycle notation. Thus if $U = \{1, 2, 3\}$, the species $\mathsf{Cyc}$ produces $\mathsf{Cyc}[U] = \{(123), (132)\}$.
- $\mathsf{Der}$ is the species of derangements, or permutations that leave no fixed points. For $U = \{1, 2, 3\}$, we have $\mathsf{Der}[U] = \{(123), (132)\}$, where the permutations have been written in cycle notation.

**Question 3.1 (2).** Let $\mathcal{I}$ be the species of involutions, or permutations $\sigma$ such that $\sigma \circ \sigma = e$. Here $e$ is the identity. Then by definition, $\mathcal{I}[U]$ is the set of involutions of $U$. List two $\mathcal{I}$ structures on $U = \{1, 2, 3, 4\}$.

**Question 3.2 (2).** Let $V_1 = \{1, 2, 3, 4, 5\}$ and $V_2 = \{a, b, c, d, e\}$. What is the difference between trees in the sets $\mathcal{T}[V_1]$ and $\mathcal{T}[V_2]$?

**Question 3.3 (3).** Let $\mathsf{Par}$ be the species of (unordered) partitions of a set. For $U = \{1, 2, 3, 4\}$, what does $\mathsf{Par}[U]$ represent? Give an example of an element of $\mathsf{Par}[U]$. Finally, compute $\left|\mathsf{Par}[U]\right|$. Justification is not required for this question.

Our ultimate goal with species, recall, is to *relate* different combinatorial species to each other in a nice way. This requires a notion of equality.

**Definition 3.1.2.** Let $\mathcal{F}$ and $\mathcal{G}$ be two species of (combinatorial) objects. Then $\mathcal{F}$ and $\mathcal{G}$ are **isomorphic**, denoted $\mathcal{F} = \mathcal{G}$, if they satisfy a condition of **naturality**: there is a bijection between $\mathcal{F}[U]$ and $\mathcal{G}[U]$ that does not depend on the specific elements of $U$.

What this definition says is that anytime we want to show that two species are *isomorphic*, it is enough to construct a bijection between $f \in \mathcal{F}[U]$ and $g \in \mathcal{G}[U]$ from the two species that does not depend on the labels from $U$. See Example 3.2.

### 3.2 Species Operations

**Definition 3.2.1** (Addition)**.** Let $\mathcal{F}$ and $\mathcal{G}$ be two combinatorial species. Then the **sum** $\mathcal{F}+\mathcal{G}$ of $\mathcal{F}$ and $\mathcal{G}$ is also a species, defined as follows:
For a set $U$, an element $s$ lies in $(\mathcal{F}+\mathcal{G})[U]$ if $s$ is a $\mathcal{F}$ structure on $U$ or a $\mathcal{G}$ structure on $U$, but not both.

**Definition 3.2.2** (Multiplication)**.** Let $\mathcal{F}$ and $\mathcal{G}$ be two combinatorial species. Then $\mathcal{F}\cdot\mathcal{G}$ (or equivalently $\mathcal{F}\mathcal{G}$), called the **product** of $\mathcal{F}$ and $\mathcal{G}$, is also a species, defined as follows:
An $\mathcal{F}\cdot\mathcal{G}$ structure on $U$, say $p \in \mathcal{F}\mathcal{G}[U]$, is given by the ordered pair $(f, g)$, where $f \in \mathcal{F}[U_1]$ and $g \in \mathcal{G}[U_2]$ for some partition $(U_1, U_2)$ of $U$ (meaning $U_1 \cup U_2 = U$ and $U_1 \cap U_2 = \emptyset$).
Thus for any set $U$, we have

$$\mathcal{F}\mathcal{G}[U] = \text{Union of } \mathcal{F}[U_1] \times \mathcal{G}[U_2] \text{ over all ways to partition } U \text{ into } (U_1, U_2).$$

It is evident that we don't want to use the previous definitions for all our purposes. The following theorem allows us to think about sums and products in a simpler way, without having to worry about the original definitions.

**Theorem 1** (Sum and Product Rules)**.** *Combinatorial sum and product theorems translate directly to sums and products of species.*

**Example 3.2.** Consider the combinatorial species $\mathcal{B}$ of (rooted) binary trees. Consider a structure $b \in \mathcal{B}[U]$, which is a (possibly empty) binary tree with labels from $U$. One case is that $b$ is empty, which corresponds to the 1 species. If $b$ is not empty, remove the root of $b$, which is the species $\mathcal{Z}$. The remaining structure will be a (possibly empty) binary tree spawning from one of the children, **and** another from the other child. So

$$b \iff \text{empty } \textbf{or } [\text{root } \textbf{and } (\text{possibly empty}) \text{ binary tree } \textbf{and } (\text{possibly empty}) \text{ binary tree}]$$

Since the bijection is independent of what elements of $U$ actually are, in terms of species, this translates into the isomorphism

$$\mathcal{B} = 1 + \mathcal{Z} \cdot (\mathcal{B}) \cdot (\mathcal{B})$$

due to the Sum and Product rules. Thus

$$\mathcal{B} = 1 + \mathcal{Z} \cdot \mathcal{B}^2.$$

.

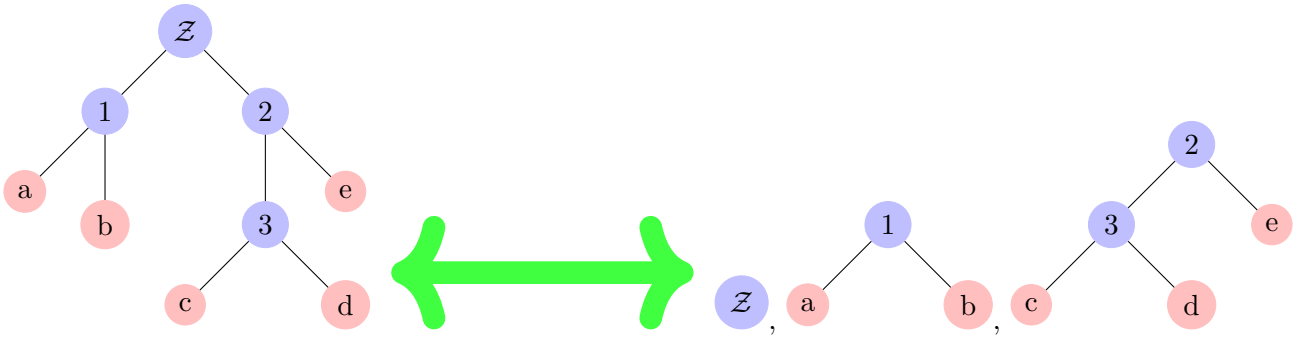


Figure 5: Bijection between binary trees, which leads to a species isomorphism (specifically, $\mathcal{B}[\{\mathcal{Z}, 1, 2, 3\}]$ is shown). Observe that the labels themselves have not been a part of our bijection.

**Note:** The expectation throughout the power round is that species isomorphism proofs are of the form of Example 3.2; that is, using Theorem 1 and bijections that are independent of labels $U$.

**Question 3.4.** Prove the following isomorphisms in the manner of Example 3.2.

(a) **(2)** Prove that $\mathsf{Per} = \mathsf{Set} \cdot \mathsf{Der}$. (*Hint:* A permutation has points that stay fixed and those that don't.)

(b) **(2)** Let $\mathcal{P}$ be the species of power sets. That is, for a finite set $U$, a structure in $\mathcal{P}[U]$ is a set of subsets of $U$. Prove that $\mathcal{P} = \mathsf{Set} \cdot \mathsf{Set}$.

The definition of composition is extremely convoluted, but it serves as a valuable operation in the theory of species. We will focus solely on **set composition** of species, since this has the most applications in the theory and is easiest to understand.

**Definition 3.2.3.** A **set composition** $\mathcal{C} = \mathsf{Set} \circ \mathcal{G}$, also denoted $\mathcal{C} = \mathsf{Set}(\mathcal{G})$, is a combinatorial species. A $\mathcal{C}$ structure on some label set $U$ is a collection (or set) of $\mathcal{G}$ structures $\{g_1, g_2, \ldots, g_k\}$, where each $g_i$ is an $\mathcal{G}$ structure defined on a set partition $U_i$. (In other words, $U_1 \cup U_2 \cup \cdots \cup U_k = U$, and $U_i \cap U_j = \emptyset$ for all $1 \leq i \neq j \leq k$.)

**Example 3.3.** Considering the species $\mathsf{Cyc}$ of cycles, if $U = \{1, 2, 3, 4\}$, then $\big\{(1234)\big\}$, $\big\{(13), (24)\big\}$, and $\big\{(1), (2), (34)\big\}$ are elements of $\mathsf{Set}(\mathsf{Cyc})[U]$.

**Question 3.5.** Prove the following relations:

(a) **(2)** Let $\mathsf{Par}$ be the species of partitions. Then $\mathsf{Par} = \mathsf{Set} \circ \mathsf{Set}_+$, where $\mathsf{Set}_+$ is the species of non-empty sets so that $1 + \mathsf{Set}_+ = \mathsf{Set}$. Recall that 1 is the one-species

(b) **(3)** $\mathsf{Per} = \mathsf{Set} \circ \mathsf{Cyc}$.

# 4 Magic

## 4.1 Combinatorial Species and Generating Functions

The past two sections are connected in a very beautiful manner that we will now get to experience. Generating functions are an amazing way to encode all the information about species. Since combinatorics is about enumeration, we concern ourselves with $\big|\mathcal{F}[U]\big|$; without loss of generality, we only consider $|\mathcal{F}[n]| := \big|\mathcal{F}[\{1, 2, \ldots, n\}]\big|$.

**Definition 4.1.1.** For a species $\mathcal{F}$, the associated exponential generating function is defined to be the formal power series

$$F(z) = \sum_{n=0}^{\infty} f_n \frac{z^n}{n!},$$

where $f_n = |\mathcal{F}[n]|$.

**Example 4.1.** Let $B(z) = \sum_{n=0}^{\infty} b_n \frac{z^n}{n!}$ be the EGF for binary trees. Then $b_n = |\mathcal{B}[n]|$ is the number of binary trees on a set with $n$ nodes.

**Example 4.2.** Consider $\mathsf{Set}$ and its generating function $\mathrm{Set}(z) = \sum_{n=0}^{\infty} s_n \frac{z^n}{n!}$. Recall that for nonempty $U$, $\mathsf{Set}[U]$ has only one element $\{U\}$, so $s_n = 1$ for all $n$. But from Section 3, we know then that

$$\mathrm{Set}(z) = \sum_{n=0}^{\infty} 1 \cdot \frac{z^n}{n!} = e^z.$$

**Question 4.1.** Prove the following EGF-species correspondences:

(a) **(2)** The EGF for $\mathcal{F} + \mathcal{G}$ is $F(z) + G(z)$.

(b) **(5)** The EGF for $\mathcal{F} \cdot \mathcal{G}$ is $F(z)G(z)$. (*Hint:* Compare the product definition 3.2.2 to the result of Problem 2.2.)

Now we can give true meaning to a lot of the species isomorphisms. This is accomplished by the following theorem.

**Theorem 2.** *Isomorphic species have the same EGF.*

The EGF-species correspondence extends to set composition in the expected way.

**Theorem 3.** *The EGF of* $\mathsf{Set} \circ \mathcal{G}$ *is* $\exp(G(z)) = e^{G(z)}$*, given that* $[z^0/0!]G(z) = 0$*. In other words, the* $z^0/0!$ *term, or constant term, is zero for* $G(z)$*.*

A proof of this theorem relies on a lot of heavy algebra and the actual definition of composition. We avoid that exposition here; however, feel free to use Theorem 3 in future exercises.

## 4.2 Permutation Statistics

Since all our work has been regarding labelled structures, one of the really amazing things that we can do is derive interesting results about permutations.
**Note:** All of the questions in this section can be solved nicely using the theory of species developed so far, but you are free to use any method of proof.

**Question 4.2 (5).** Let $\mathcal{I}$ be the species of involutions. Show that $\mathcal{I}$ has EGF

$$I(z) = \exp\left(z + \frac{z^2}{2}\right).$$

Use this to show that we have the formula

$$I_n = \left[\frac{z^n}{n!}\right] I(z) = \sum_{k=0}^{\lfloor n/2 \rfloor} \binom{n}{2k} \frac{(2k)!}{k! \cdot 2^k}.$$

*Hint:* Try to prove the species isomorphism $\mathcal{I} = \mathsf{Set}(\mathsf{Cyc}_1 + \mathsf{Cyc}_2)$, where $\mathsf{Cyc}_j$ denotes the Species of cycles that are of length $k$. This says that involutions can only have cycles of length 1 and 2. Why?

**Question 4.3 (5).** Generalize the result of Question 4.2: let $\mathcal{U}_m$ be the species of "$m$th roots of unity," or permutations $\sigma$ such that $\underbrace{\sigma \circ \sigma \circ \sigma \cdots \circ \sigma}_{m \text{ times}} = \sigma^m =$ identity. Show that the EGF $U_m(z)$ of $\mathcal{U}_m$ is

$$U_m(z) = \exp\left(\sum_{d|m} \frac{z^d}{d}\right).$$

**Question 4.4 (8).** Let $\mathbb{P}_k(n)$ be the probability that a random permutation of $[n]$ does **not** contain a cycle of length $k$. Show that $\mathbb{P}_k(n) \to e^{-1/k}$ as $n \to \infty$.
*Hint:* Let $P_k(n)$ be the number of permutations of $[n]$ without a cycle of length $k$, and set $P_k(z) = \sum_{n=0}^{\infty} P_k(n) \frac{z^n}{n!}$. Argue why evaluating the telescoping sum $(1-z)P_k(z)$ at $z = 1$ allows us to compute $\lim_{n\to\infty} \mathbb{P}_k(n) = \lim_{n\to\infty} \frac{P_k(n)}{n!}$.

**Question 4.5 (10).** Show that the probability that the shortest cycle of a permutation of $[n]$ has length greater than $k$ as $n \to \infty$ is

$$\approx \frac{1}{e^\gamma k}.$$

The $\approx$ symbol comes from

$$1 + \frac{1}{2} + \cdots + \frac{1}{k} \approx \ln(k) + \gamma,$$

where $\gamma$ is the Euler-Mascheroni constant; feel free to treat this approximation as equality.
*Hint:* Start with $\mathsf{Per}_{>k} = \mathsf{Set} \circ \mathsf{Cyc}_{>k}$. Here $\mathsf{Per}_{>k}$ is the species of permutations with shortest cycle of length greater than $k$. $\mathsf{Cyc}_{>k}$ is the species of cycles of length greater than $k$.