AMT 223 1.0 Discrete Mathematics (General Degree)

Dr. G.H.J. Lanel

Semester 2 - 2018

Outline

1. Modeling Combinatorial Problems with Recurrence Relations
   - Recurrence Relation not in Closed Form
   - Analysis of Algorithms

- We previously showed sequences can be defined recursively.
- Indeed, some sequences have no simple definition other than a recursive one.
- In this section we look at sequences that are not defined recursively (they may be defined in terms of an application) but for which a recursive formula can be written down.

- We previously showed sequences can be defined recursively.
- Indeed, some sequences have no simple definition other than a recursive one.
- In this section we look at sequences that are not defined recursively (they may be defined in terms of an application) but for which a recursive formula can be written down.

- We previously showed sequences can be defined recursively.
- Indeed, some sequences have no simple definition other than a recursive one.
- In this section we look at sequences that are not defined recursively (they may be defined in terms of an application) but for which a recursive formula can be written down.

- Such a formula is called a **recurrence relation** for the sequence.
- The advantage of such a formula is twofold.
- First, it allows us to compute the terms in the sequence, one at a time.
- Second, as we will see in future, it sometimes allows us to derive a closed-form, non-recursive formula for the terms in the sequence.
- Throughout our discussion, the expression defining f can also involve n, even though we do not show it explicitly as an argument.

- Such a formula is called a **recurrence relation** for the sequence.
- The advantage of such a formula is twofold.
- First, it allows us to compute the terms in the sequence, one at a time.
- Second, as we will see in future, it sometimes allows us to derive a closed-form, non-recursive formula for the terms in the sequence.
- Throughout our discussion, the expression defining f can also involve n, even though we do not show it explicitly as an argument.

- Such a formula is called a **recurrence relation** for the sequence.
- The advantage of such a formula is twofold.
- First, it allows us to compute the terms in the sequence, one at a time.
- Second, as we will see in future, it sometimes allows us to derive a closed-form, non-recursive formula for the terms in the sequence.
- Throughout our discussion, the expression defining f can also involve n, even though we do not show it explicitly as an argument.

- Such a formula is called a **recurrence relation** for the sequence.
- The advantage of such a formula is twofold.
- First, it allows us to compute the terms in the sequence, one at a time.
- Second, as we will see in future, it sometimes allows us to derive a closed-form, non-recursive formula for the terms in the sequence.
- Throughout our discussion, the expression defining f can also involve n, even though we do not show it explicitly as an argument.

- Such a formula is called a **recurrence relation** for the sequence.
- The advantage of such a formula is twofold.
- First, it allows us to compute the terms in the sequence, one at a time.
- Second, as we will see in future, it sometimes allows us to derive a closed-form, non-recursive formula for the terms in the sequence.
- Throughout our discussion, the expression defining f can also involve n, even though we do not show it explicitly as an argument.

Examples

- $a_n = na_{n-1}^2 + a_3 + 5$ is a recurrence relation.
- Often only the immediately preceding term of the sequence enters into the recurrence relation, so that $a_n = f(a_{n-1})$ for all $n \geqslant 1$; in this case we say that we have a first-order recurrence relation.
- $a_n = \frac{n^2+1}{a_{n-1}}$ is a first order recurrence relation.

Examples

- $a_n = na_{n-1}^2 + a_3 + 5$ is a recurrence relation.
- Often only the immediately preceding term of the sequence enters into the recurrence relation, so that $a_n = f(a_{n-1})$ for all $n \geqslant 1$; in this case we say that we have a first-order recurrence relation.
- $a_n = \frac{n^2+1}{a_{n-1}}$ is a first order recurrence relation.

Examples

- $a_n = na_{n-1}^2 + a_3 + 5$ is a recurrence relation.
- Often only the immediately preceding term of the sequence enters into the recurrence relation, so that $a_n = f(a_{n-1})$ for all $n \geqslant 1$; in this case we say that we have a first-order recurrence relation.
- $a_n = \frac{n^2+1}{a_{n-1}}$ is a first order recurrence relation.

- More generally, if we have $a_n = f(a_{n-1}, a_{n-2}, ..., a_{n-k})$ for all $n \geqslant k$, then the recurrence relation is said to be of **order** k.
- Often, the restriction $n \geqslant k$ is not written explicitly, but it is to be understood nonetheless, since if $n < k$, then the term a_{n-k} would make no sense.

- More generally, if we have $a_n = f(a_{n-1}, a_{n-2}, ..., a_{n-k})$ for all $n \geqslant k$, then the recurrence relation is said to be of **order** k.
- Often, the restriction $n \geqslant k$ is not written explicitly, but it is to be understood nonetheless, since if $n < k$, then the term a_{n-k} would make no sense.

- Given a sequence that satisfies a kth-order recurrence relation, together with specific values for $a_0, a_1, \ldots, a_{k-1}$, we can write down as many terms of the sequence as we wish.
- The specifications of the values of a_0 through a_{k-1} are called initial conditions.
- Occasionally, the sequence begins at an index other than 0.
- For example,
- we may have a kth-order recurrence relation valid for all $n > k$ with initial conditions specifying $a_1, a_2, \ldots, a_k$

- Given a sequence that satisfies a kth-order recurrence relation, together with specific values for $a_0, a_1, \ldots, a_{k-1}$, we can write down as many terms of the sequence as we wish.
- The specifications of the values of a_0 through a_{k-1} are called initial conditions.
- Occasionally, the sequence begins at an index other than 0.
- For example,
- we may have a kth-order recurrence relation valid for all $n > k$ with initial conditions specifying $a_1, a_2, \ldots, a_k$

- Given a sequence that satisfies a kth-order recurrence relation, together with specific values for $a_0, a_1, \ldots, a_{k-1}$, we can write down as many terms of the sequence as we wish.
- The specifications of the values of a_0 through a_{k-1} are called initial conditions.
- Occasionally, the sequence begins at an index other than 0.
- For example,
- we may have a kth-order recurrence relation valid for all $n > k$ with initial conditions specifying $a_1, a_2, \ldots, a_k$

- Given a sequence that satisfies a kth-order recurrence relation, together with specific values for $a_0, a_1, \ldots, a_{k-1}$, we can write down as many terms of the sequence as we wish.
- The specifications of the values of a_0 through a_{k-1} are called initial conditions.
- Occasionally, the sequence begins at an index other than 0.
- For example,
- we may have a kth-order recurrence relation valid for all $n > k$ with initial conditions specifying $a_1, a_2, \ldots, a_k$

- Given a sequence that satisfies a kth-order recurrence relation, together with specific values for a_0,a_1, ... ,a_{k-1}, we can write down as many terms of the sequence as we wish.
- The specifications of the values of a_0 through a_{k-1} are called initial conditions.
- Occasionally, the sequence begins at an index other than 0.
- For example,
- we may have a kth-order recurrence relation valid for all $n > k$ with initial conditions specifying a_1,a_2, ... ,a_k

Example 1

- We saw in a previous section that the Fibonacci sequence f_n satisfies the second-order recurrence relation $f_n = f_{n-1} + f_{n-2}$ for all $n \geqslant 2$, together with the initial conditions $f_0 = 1$ and $f_1 = 1$.
- Indeed, in this case the recurrence relation and initial conditions form the definition of the sequence.
- Knowing the initial conditions and recurrence relation, we can compute the terms of the sequence, one by one.
- In this case, we find that

$$f_2 = f_1 + f_0 = 1 + 1 = 2$$

$$f_3 = f_2 + f_1 = 2 + 1 = 3$$

$$f_4 = f_3 + f_2 = 3 + 2 = 5$$

and so on.

Example 1

- We saw in a previous section that the Fibonacci sequence f_n satisfies the second-order recurrence relation $f_n = f_{n-1} + f_{n-2}$ for all $n \geqslant 2$, together with the initial conditions $f_0 = 1$ and $f_1 = 1$.
- Indeed, in this case the recurrence relation and initial conditions form the definition of the sequence.
- Knowing the initial conditions and recurrence relation, we can compute the terms of the sequence, one by one.
- In this case, we find that

$$f_2 = f_1 + f_0 = 1 + 1 = 2$$

$$f_3 = f_2 + f_1 = 2 + 1 = 3$$

$$f_4 = f_3 + f_2 = 3 + 2 = 5$$

and so on.

Example 1

- We saw in a previous section that the Fibonacci sequence f_n satisfies the second-order recurrence relation $f_n = f_{n-1} + f_{n-2}$ for all $n \geqslant 2$, together with the initial conditions $f_0 = 1$ and $f_1 = 1$.
- Indeed, in this case the recurrence relation and initial conditions form the definition of the sequence.
- Knowing the initial conditions and recurrence relation, we can compute the terms of the sequence, one by one.
- In this case, we find that

$$f_2 = f_1 + f_0 = 1 + 1 = 2$$

$$f_3 = f_2 + f_1 = 2 + 1 = 3$$

$$f_4 = f_3 + f_2 = 3 + 2 = 5$$

and so on.

Example 1

- We saw in a previous section that the Fibonacci sequence f_n satisfies the second-order recurrence relation $f_n = f_{n-1} + f_{n-2}$ for all $n \geqslant 2$, together with the initial conditions $f_0 = 1$ and $f_1 = 1$.
- Indeed, in this case the recurrence relation and initial conditions form the definition of the sequence.
- Knowing the initial conditions and recurrence relation, we can compute the terms of the sequence, one by one.
- In this case, we find that

$$f_2 = f_1 + f_0 = 1 + 1 = 2$$

$$f_3 = f_2 + f_1 = 2 + 1 = 3$$

$$f_4 = f_3 + f_2 = 3 + 2 = 5$$

and so on.

Example 2

- The sequence given by the explicit formula $a_n = n(n-1)/2$ satisfies the first-order recurrence relation $a_n = a_{n-1} + (n-1)$ for all $n \geqslant 1$, since we have

$$a_{n-1} + (n-1) = \frac{(n-1)(n-2)}{2} + (n-1) = (n-1).\frac{n}{2} = a_n$$

- The initial condition here is that $a_0 = 0$.

Example 2

- The sequence given by the explicit formula $a_n = n(n-1)/2$ satisfies the first-order recurrence relation $a_n = a_{n-1} + (n-1)$ for all $n \geqslant 1$, since we have

$$a_{n-1} + (n-1) = \frac{(n-1)(n-2)}{2} + (n-1) = (n-1).\frac{n}{2} = a_n$$

- The initial condition here is that $a_0 = 0$.

Example 2

- The sequence given by the explicit formula $a_n = n(n-1)/2$ satisfies the first-order recurrence relation $a_n = a_{n-1} + (n-1)$ for all $n \geqslant 1$, since we have

$$a_{n-1} + (n-1) = \frac{(n-1)(n-2)}{2} + (n-1) = (n-1).\frac{n}{2} = a_n$$

- The initial condition here is that $a_0 = 0$.

Example 3

- The sequence 1, 2, 4, 8, 16, ... satisfies the recurrence relation $a_n = a_{n-1} + a_{n-2} + ... + a_1 + a_0 + 1$.
- In other words, each term in this sequence is the sum of all the previous terms, plus 1.
- It also satisfies other recurrence relations, such as the first order relation $a_n = 2a_{n-1}$ and the third order relation $a_n = a_{n-1} + 4a_{n-3}$.

Example 3

- The sequence 1, 2, 4, 8, 16, ... satisfies the recurrence relation $a_n = a_{n-1} + a_{n-2} + ... + a_1 + a_0 + 1$.
- In other words, each term in this sequence is the sum of all the previous terms, plus 1.
- It also satisfies other recurrence relations, such as the first order relation $a_n = 2a_{n-1}$ and the third order relation $a_n = a_{n-1} + 4a_{n-3}$.

Example 3

- The sequence 1, 2, 4, 8, 16, ... satisfies the recurrence relation $a_n = a_{n-1} + a_{n-2} + ... + a_1 + a_0 + 1$.
- In other words, each term in this sequence is the sum of all the previous terms, plus 1.
- It also satisfies other recurrence relations, such as the first order relation $a_n = 2a_{n-1}$ and the third order relation $a_n = a_{n-1} + 4a_{n-3}$.

- Our main goal in this section is to set up recurrence relations for solving problems.
- Typically, we are given a sequence defined in concrete, not algebraic, terms; we want to write down a recurrence relation that the sequence must satisfy.

- Our main goal in this section is to set up recurrence relations for solving problems.
- Typically, we are given a sequence defined in concrete, not algebraic, terms; we want to write down a recurrence relation that the sequence must satisfy.

Example 4

Let b_n be the number of bit strings of length n containing a pair of consecutive 0's. Find a recurrence relation and initial conditions for the sequence b_n.

Solution

- There are three mutually exclusive ways that such a sequence might start: 1, 01, and 00.
- If it starts with a 1, it must continue with a bit string of length $n-1$ containing a pair of consecutive 0's, and there are b_{n-1} of these.
- If it starts with 01, it must continue with a bit string of length $n-2$ containing a pair of consecutive 0's, and there are b_{n-2} of these.

Example 4

Let b_n be the number of bit strings of length n containing a pair of consecutive 0's. Find a recurrence relation and initial conditions for the sequence b_n.

Solution

- There are three mutually exclusive ways that such a sequence might start: 1, 01, and 00.
- If it starts with a 1, it must continue with a bit string of length $n-1$ containing a pair of consecutive 0's, and there are b_{n-1} of these.
- If it starts with 01, it must continue with a bit string of length $n-2$ containing a pair of consecutive 0's, and there are b_{n-2} of these.

Example 4

Let b_n be the number of bit strings of length n containing a pair of consecutive 0's. Find a recurrence relation and initial conditions for the sequence b_n.

Solution

- There are three mutually exclusive ways that such a sequence might start: 1, 01, and 00.
- If it starts with a 1, it must continue with a bit string of length $n-1$ containing a pair of consecutive 0's, and there are b_{n-1} of these.
- If it starts with 01, it must continue with a bit string of length $n-2$ containing a pair of consecutive 0's, and there are b_{n-2} of these.

Example 4

Let b_n be the number of bit strings of length n containing a pair of consecutive 0's. Find a recurrence relation and initial conditions for the sequence b_n.

Solution

- There are three mutually exclusive ways that such a sequence might start: 1, 01, and 00.
- If it starts with a 1, it must continue with a bit string of length $n-1$ containing a pair of consecutive 0's, and there are b_{n-1} of these.
- If it starts with 01, it must continue with a bit string of length $n-2$ containing a pair of consecutive 0's, and there are b_{n-2} of these.

Example 4

Let b_n be the number of bit strings of length n containing a pair of consecutive 0's. Find a recurrence relation and initial conditions for the sequence b_n.

Solution

- There are three mutually exclusive ways that such a sequence might start: 1, 01, and 00.
- If it starts with a 1, it must continue with a bit string of length $n-1$ containing a pair of consecutive 0's, and there are b_{n-1} of these.
- If it starts with 01, it must continue with a bit string of length $n-2$ containing a pair of consecutive 0's, and there are b_{n-2} of these.

Example 4

- Finally, if it starts 00, it can be followed by any bit string of length $n-2$ (since a pair of consecutive 0's is already present), and there are 2^{n-2} of these.
- Therefore, the desired recurrence relation is $b_n = b_{n-1} + b_{n-2} + 2^{n-2}$.
- Clearly, the initial conditions are $b_0 = b_1 = 0$, since no strings of length less than 2 can contain 00 as a substring.

Example 4

- Finally, if it starts 00, it can be followed by any bit string of length $n-2$ (since a pair of consecutive 0's is already present), and there are 2^{n-2} of these.
- Therefore, the desired recurrence relation is $b_n = b_{n-1} + b_{n-2} + 2^{n-2}$.
- Clearly, the initial conditions are $b_0 = b_1 = 0$, since no strings of length less than 2 can contain 00 as a substring.

Example 4

- Finally, if it starts 00, it can be followed by any bit string of length $n-2$ (since a pair of consecutive 0's is already present), and there are 2^{n-2} of these.
- Therefore, the desired recurrence relation is $b_n = b_{n-1} + b_{n-2} + 2^{n-2}$.
- Clearly, the initial conditions are $b_0 = b_1 = 0$, since no strings of length less than 2 can contain 00 as a substring.

Example 4

- With this recurrence relation, we can compute the terms in the sequence.
- We have

$b_2 = b_1 + b_0 + 2^0 = 0 + 0 + 1 = 1$

$b_3 = b_2 + b_1 + 2^1 = 1 + 0 + 2 = 3$

$b_4 = b_3 + b_2 + 2^2 = 3 + 1 + 4 = 8$

$b_5 = b_4 + b_3 + 2^3 = 8 + 3 + 8 = 19$

$b_t 6 = b_5 + b_4 + 2^4 = 19 + 8 + 16 = 43$

$b_7 = b_6 + b_5 + 2^5 = 43 + 19 + 32 = 94$

$b_8 = b_7 + b_6 + 2^6 = 94 + 43 + 64 = 201$

and so on.

Example 4

- With this recurrence relation, we can compute the terms in the sequence.
- We have

$$b_2 = b_1 + b_0 + 2^0 = 0 + 0 + 1 = 1$$

$$b_3 = b_2 + b_1 + 2^1 = 1 + 0 + 2 = 3$$

$$b_4 = b_3 + b_2 + 2^2 = 3 + 1 + 4 = 8$$

$$b_5 = b_4 + b_3 + 2^3 = 8 + 3 + 8 = 19$$

$$b_t 6 = b_5 + b_4 + 2^4 = 19 + 8 + 16 = 43$$

$$b_7 = b_6 + b_5 + 2^5 = 43 + 19 + 32 = 94$$

$$b_8 = b_7 + b_6 + 2^6 = 94 + 43 + 64 = 201$$

and so on.

Example 4

- With this recurrence relation, we can compute the terms in the sequence.
- We have

$$b_2 = b_1 + b_0 + 2^0 = 0 + 0 + 1 = 1$$

$$b_3 = b_2 + b_1 + 2^1 = 1 + 0 + 2 = 3$$

$$b_4 = b_3 + b_2 + 2^2 = 3 + 1 + 4 = 8$$

$$b_5 = b_4 + b_3 + 2^3 = 8 + 3 + 8 = 19$$

$$b_t 6 = b_5 + b_4 + 2^4 = 19 + 8 + 16 = 43$$

$$b_7 = b_6 + b_5 + 2^5 = 43 + 19 + 32 = 94$$

$$b_8 = b_7 + b_6 + 2^6 = 94 + 43 + 64 = 201$$

and so on.

Example 4

- With this recurrence relation, we can compute the terms in the sequence.
- We have

$$b_2 = b_1 + b_0 + 2^0 = 0 + 0 + 1 = 1$$

$$b_3 = b_2 + b_1 + 2^1 = 1 + 0 + 2 = 3$$

$$b_4 = b_3 + b_2 + 2^2 = 3 + 1 + 4 = 8$$

$$b_5 = b_4 + b_3 + 2^3 = 8 + 3 + 8 = 19$$

$$b_t 6 = b_5 + b_4 + 2^4 = 19 + 8 + 16 = 43$$

$$b_7 = b_6 + b_5 + 2^5 = 43 + 19 + 32 = 94$$

$$b_8 = b_7 + b_6 + 2^6 = 94 + 43 + 64 = 201$$

and so on.

Example 4

- With this recurrence relation, we can compute the terms in the sequence.
- We have

$$b_2 = b_1 + b_0 + 2^0 = 0 + 0 + 1 = 1$$

$$b_3 = b_2 + b_1 + 2^1 = 1 + 0 + 2 = 3$$

$$b_4 = b_3 + b_2 + 2^2 = 3 + 1 + 4 = 8$$

$$b_5 = b_4 + b_3 + 2^3 = 8 + 3 + 8 = 19$$

$$b_t 6 = b_5 + b_4 + 2^4 = 19 + 8 + 16 = 43$$

$$b_7 = b_6 + b_5 + 2^5 = 43 + 19 + 32 = 94$$

$$b_8 = b_7 + b_6 + 2^6 = 94 + 43 + 64 = 201$$

and so on.

Example 4

- With this recurrence relation, we can compute the terms in the sequence.
- We have

$$b_2 = b_1 + b_0 + 2^0 = 0 + 0 + 1 = 1$$

$$b_3 = b_2 + b_1 + 2^1 = 1 + 0 + 2 = 3$$

$$b_4 = b_3 + b_2 + 2^2 = 3 + 1 + 4 = 8$$

$$b_5 = b_4 + b_3 + 2^3 = 8 + 3 + 8 = 19$$

$$b_t 6 = b_5 + b_4 + 2^4 = 19 + 8 + 16 = 43$$

$$b_7 = b_6 + b_5 + 2^5 = 43 + 19 + 32 = 94$$

$$b_8 = b_7 + b_6 + 2^6 = 94 + 43 + 64 = 201$$

and so on.

Example 4

- With this recurrence relation, we can compute the terms in the sequence.
- We have

$$b_2 = b_1 + b_0 + 2^0 = 0 + 0 + 1 = 1$$

$$b_3 = b_2 + b_1 + 2^1 = 1 + 0 + 2 = 3$$

$$b_4 = b_3 + b_2 + 2^2 = 3 + 1 + 4 = 8$$

$$b_5 = b_4 + b_3 + 2^3 = 8 + 3 + 8 = 19$$

$$b_t 6 = b_5 + b_4 + 2^4 = 19 + 8 + 16 = 43$$

$$b_7 = b_6 + b_5 + 2^5 = 43 + 19 + 32 = 94$$

$$b_8 = b_7 + b_6 + 2^6 = 94 + 43 + 64 = 201$$

and so on.

Example 4

- With this recurrence relation, we can compute the terms in the sequence.
- We have

$$b_2 = b_1 + b_0 + 2^0 = 0 + 0 + 1 = 1$$

$$b_3 = b_2 + b_1 + 2^1 = 1 + 0 + 2 = 3$$

$$b_4 = b_3 + b_2 + 2^2 = 3 + 1 + 4 = 8$$

$$b_5 = b_4 + b_3 + 2^3 = 8 + 3 + 8 = 19$$

$$b_t 6 = b_5 + b_4 + 2^4 = 19 + 8 + 16 = 43$$

$$b_7 = b_6 + b_5 + 2^5 = 43 + 19 + 32 = 94$$

$$b_8 = b_7 + b_6 + 2^6 = 94 + 43 + 64 = 201$$

and so on.

- Unfortunately, there is no algorithm to tell us how to analyze an applied problem, such as the ones we have been considering here, to come up with a recurrence relation.
- A successful analysis often takes a bit of cleverness and usually involves one or more false starts.
- In the rest of this section we turn to problems that are somewhat more involved than the ones we have looked at so far.

- Unfortunately, there is no algorithm to tell us how to analyze an applied problem, such as the ones we have been considering here, to come up with a recurrence relation.
- A successful analysis often takes a bit of cleverness and usually involves one or more false starts.
- In the rest of this section we turn to problems that are somewhat more involved than the ones we have looked at so far.

- Unfortunately, there is no algorithm to tell us how to analyze an applied problem, such as the ones we have been considering here, to come up with a recurrence relation.
- A successful analysis often takes a bit of cleverness and usually involves one or more false starts.
- In the rest of this section we turn to problems that are somewhat more involved than the ones we have looked at so far.

Recurrence Relation not in Closed Form

- Let $p(n)$ be the number of partitions of a set with n elements. ($p(n)$ is also the number of different equivalence relations on a set with n elements.)
- The numbers $p(n)$ are known as the Bell numbers, after the American mathematician E. T. Bell.
- For example, $p(3) = 5$, since the partitions of $\{1, 2, 3\}$ are $\{\{1, 2, 3\}\}$,$\{\{1, 2\}, \{3\}\}$,$\{\{1, 3\}, \{2\}\}$,$\{\{2, 3\}, \{1\}\}$, and $\{\{1\}, \{2\}, \{3\}\}$. In order to get a recurrence relation for $p(n)$, we need to see how partitions of smaller sets help to determine partitions of larger ones.

Recurrence Relation not in Closed Form

- Let $p(n)$ be the number of partitions of a set with n elements. ($p(n)$ is also the number of different equivalence relations on a set with n elements.)
- The numbers $p(n)$ are known as the Bell numbers, after the American mathematician E. T. Bell.
- For example, $p(3) = 5$, since the partitions of $\{1, 2, 3\}$ are $\{\{1, 2, 3\}\}$,$\{\{1, 2\}, \{3\}\}$,$\{\{1, 3\}, \{2\}\}$,$\{\{2, 3\}, \{1\}\}$, and $\{\{1\}, \{2\}, \{3\}\}$. In order to get a recurrence relation for $p(n)$, we need to see how partitions of smaller sets help to determine partitions of larger ones.

Recurrence Relation not in Closed Form

- Let $p(n)$ be the number of partitions of a set with n elements. ($p(n)$ is also the number of different equivalence relations on a set with n elements.)
- The numbers $p(n)$ are known as the Bell numbers, after the American mathematician E. T. Bell.
- For example, $p(3) = 5$, since the partitions of $\{1, 2, 3\}$ are $\{\{1, 2, 3\}\}$,$\{\{1, 2\}, \{3\}\}$,$\{\{1, 3\}, \{2\}\}$,$\{\{2, 3\}, \{1\}\}$, and $\{\{1\}, \{2\}, \{3\}\}$. In order to get a recurrence relation for $p(n)$, we need to see how partitions of smaller sets help to determine partitions of larger ones.

- We count the partitions of $\{1, 2, .., n\}$ as follows.
- The element n must be in one of the sets of the partition.
- It can be in a set by itself, or it can have one or more (possibly even all) of the other elements of $\{1, 2, , n\}$ with it.
- Let k be the number of elements other than n in the same set with n in a partition of $\{1, 2, ..., n\}$.
- For example, if $n = 3$, then the partition $\{\{2, 3\}, \{1\}\}$ has $k = 1$ since only 2 is in the same set as 3.

- We count the partitions of $\{1, 2, .., n\}$ as follows.
- The element n must be in one of the sets of the partition.
- It can be in a set by itself, or it can have one or more (possibly even all) of the other elements of $\{1, 2, , n\}$ with it.
- Let k be the number of elements other than n in the same set with n in a partition of $\{1, 2, ..., n\}$.
- For example, if $n = 3$, then the partition $\{\{2, 3\}, \{1\}\}$ has $k = 1$ since only 2 is in the same set as 3.

- We count the partitions of $\{1, 2, .., n\}$ as follows.
- The element n must be in one of the sets of the partition.
- It can be in a set by itself, or it can have one or more (possibly even all) of the other elements of $\{1, 2, , n\}$ with it.
- Let k be the number of elements other than n in the same set with n in a partition of $\{1, 2, ..., n\}$.
- For example, if $n = 3$, then the partition $\{\{2, 3\}, \{1\}\}$ has $k = 1$ since only 2 is in the same set as 3.

- We count the partitions of $\{1, 2, .., n\}$ as follows.
- The element n must be in one of the sets of the partition.
- It can be in a set by itself, or it can have one or more (possibly even all) of the other elements of $\{1, 2, , n\}$ with it.
- Let k be the number of elements other than n in the same set with n in a partition of $\{1, 2, ..., n\}$.
- For example, if $n = 3$, then the partition $\{\{2, 3\}, \{1\}\}$ has $k = 1$ since only 2 is in the same set as 3.

- We count the partitions of $\{1, 2, .., n\}$ as follows.
- The element n must be in one of the sets of the partition.
- It can be in a set by itself, or it can have one or more (possibly even all) of the other elements of $\{1, 2, , n\}$ with it.
- Let k be the number of elements other than n in the same set with n in a partition of $\{1, 2, ..., n\}$.
- For example, if $n = 3$, then the partition $\{\{2, 3\}, \{1\}\}$ has $k = 1$ since only 2 is in the same set as 3.

- Note that $1 \leqslant k \leqslant n-1$.
- In order to specify a partition with this value of k, we can first decide which k elements are to be in the same set as n (and we can do this in $C(n-1,k)$ ways), and then decide how to partition the remaining elements of $\{1,2,...,n\}$ (and we can do this in $p(n-k-1)$ ways, since there are $n-k-1$ elements left to be partitioned).
- Therefore, by the multiplication principle there are $C(n-1,k)p(n-k-1)$ partitions of $\{1,2.,..n\}$ in which exactly k elements are in the same set as n.

- Note that $1 \leqslant k \leqslant n-1$.
- In order to specify a partition with this value of k, we can first decide which k elements are to be in the same set as n (and we can do this in $C(n-1,k)$ ways), and then decide how to partition the remaining elements of $\{1,2,...,n\}$ (and we can do this in $p(n-k-1)$ ways, since there are $n-k-1$ elements left to be partitioned).
- Therefore, by the multiplication principle there are $C(n-1,k)p(n-k-1)$ partitions of $\{1,2.,..n\}$ in which exactly k elements are in the same set as n.

- Note that $1 \leqslant k \leqslant n-1$.
- In order to specify a partition with this value of k, we can first decide which k elements are to be in the same set as n (and we can do this in $C(n-1,k)$ ways), and then decide how to partition the remaining elements of $\{1,2,...,n\}$ (and we can do this in $p(n-k-1)$ ways, since there are $n-k-1$ elements left to be partitioned).
- Therefore, by the multiplication principle there are $C(n-1,k)p(n-k-1)$ partitions of $\{1,2.,..n\}$ in which exactly k elements are in the same set as n.

- Finally, by the addition principle, the total number of partitions of $\{1, 2, ...n\}$ is given by

$$P(n) = \sum_{k=0}^{n-1} C(n-1, k)p(n-k-1).$$

- This formula is our recurrence relation; it specifies $p(n)$ in terms of the numbers $p(n-k-1)$, all of which have arguments smaller than n (since $k \geqslant 0$). The only initial condition needed is $p(0) = 1$, reflecting the fact that the empty set is the only partition of the empty set.

- Note that this recurrence relation is not of a fixed order, as were most of the recurrence relations we considered earlier in this section.

- Instead, the recurrence relation expresses $p(n)$ as a function of all the numbers $p(0)$, $p(1)$,...,$p(n-1)$ (as well as n).

- Finally, by the addition principle, the total number of partitions of $\{1, 2, ...n\}$ is given by

$$P(n) = \sum_{k=0}^{n-1} C(n-1, k)p(n-k-1).$$

- This formula is our recurrence relation; it specifies $p(n)$ in terms of the numbers $p(n-k-1)$, all of which have arguments smaller than n (since $k \geqslant 0$). The only initial condition needed is $p(0) = 1$, reflecting the fact that the empty set is the only partition of the empty set.

- Note that this recurrence relation is not of a fixed order, as were most of the recurrence relations we considered earlier in this section.

- Instead, the recurrence relation expresses $p(n)$ as a function of all the numbers $p(0)$, $p(1)$,...,$p(n-1)$ (as well as n).

- Finally, by the addition principle, the total number of partitions of $\{1, 2, ...n\}$ is given by

$$P(n) = \sum_{k=0}^{n-1} C(n-1, k)p(n-k-1).$$

- This formula is our recurrence relation; it specifies $p(n)$ in terms of the numbers $p(n-k-1)$, all of which have arguments smaller than n (since $k \geqslant 0$). The only initial condition needed is $p(0) = 1$, reflecting the fact that the empty set is the only partition of the empty set.

- Note that this recurrence relation is not of a fixed order, as were most of the recurrence relations we considered earlier in this section.

- Instead, the recurrence relation expresses $p(n)$ as a function of all the numbers $p(0)$, $p(1)$,...,$p(n-1)$ (as well as n).

- Finally, by the addition principle, the total number of partitions of $\{1, 2, ...n\}$ is given by

$$P(n) = \sum_{k=0}^{n-1} C(n-1, k)p(n-k-1).$$

- This formula is our recurrence relation; it specifies $p(n)$ in terms of the numbers $p(n-k-1)$, all of which have arguments smaller than n (since $k \geqslant 0$). The only initial condition needed is $p(0) = 1$, reflecting the fact that the empty set is the only partition of the empty set.

- Note that this recurrence relation is not of a fixed order, as were most of the recurrence relations we considered earlier in this section.

- Instead, the recurrence relation expresses $p(n)$ as a function of all the numbers $p(0)$, $p(1)$,...,$p(n-1)$ (as well as n).

Example

Find the number of partitions of a set with five elements.

Solution

- We need to compute $p(5)$, and we can do so if we first compute $p(0)$, $p(1)$, $p(2)$, $p(3)$, and $p(4)$.
- The first three of these we may as well do directly, since they are so simple.
- We already noted that $p(0) = 1$, and it is clear that $p(1) = 1$ as well.
- Also, $p(2) = 2$, since we can put the two elements either in one set together or in separate sets.
- Similarly $p(3) = 5$ several paragraphs above.

Example

Find the number of partitions of a set with five elements.

Solution

- We need to compute $p(5)$, and we can do so if we first compute $p(0)$, $p(1)$, $p(2)$, $p(3)$, and $p(4)$.
- The first three of these we may as well do directly, since they are so simple.
- We already noted that $p(0) = 1$, and it is clear that $p(1) = 1$ as well.
- Also, $p(2) = 2$, since we can put the two elements either in one set together or in separate sets.
- Similarly $p(3) = 5$ several paragraphs above.

Example

Find the number of partitions of a set with five elements.

Solution

- We need to compute $p(5)$, and we can do so if we first compute $p(0)$, $p(1)$, $p(2)$, $p(3)$, and $p(4)$.
- The first three of these we may as well do directly, since they are so simple.
- We already noted that $p(0) = 1$, and it is clear that $p(1) = 1$ as well.
- Also, $p(2) = 2$, since we can put the two elements either in one set together or in separate sets.
- Similarly $p(3) = 5$ several paragraphs above.

Example

Find the number of partitions of a set with five elements.

Solution

- We need to compute $p(5)$, and we can do so if we first compute $p(0)$, $p(1)$, $p(2)$, $p(3)$, and $p(4)$.
- The first three of these we may as well do directly, since they are so simple.
- We already noted that $p(0) = 1$, and it is clear that $p(1) = 1$ as well.
- Also, $p(2) = 2$, since we can put the two elements either in one set together or in separate sets.
- Similarly $p(3) = 5$ several paragraphs above.

Example

Find the number of partitions of a set with five elements.

Solution

- We need to compute $p(5)$, and we can do so if we first compute $p(0)$, $p(1)$, $p(2)$, $p(3)$, and $p(4)$.
- The first three of these we may as well do directly, since they are so simple.
- We already noted that $p(0) = 1$, and it is clear that $p(1) = 1$ as well.
- Also, $p(2) = 2$, since we can put the two elements either in one set together or in separate sets.
- Similarly $p(3) = 5$ several paragraphs above.

Example

Find the number of partitions of a set with five elements.

Solution

- We need to compute $p(5)$, and we can do so if we first compute $p(0)$, $p(1)$, $p(2)$, $p(3)$, and $p(4)$.
- The first three of these we may as well do directly, since they are so simple.
- We already noted that $p(0) = 1$, and it is clear that $p(1) = 1$ as well.
- Also, $p(2) = 2$, since we can put the two elements either in one set together or in separate sets.
- Similarly $p(3) = 5$ several paragraphs above.

Example

Find the number of partitions of a set with five elements.

Solution

- We need to compute $p(5)$, and we can do so if we first compute $p(0)$, $p(1)$, $p(2)$, $p(3)$, and $p(4)$.
- The first three of these we may as well do directly, since they are so simple.
- We already noted that $p(0) = 1$, and it is clear that $p(1) = 1$ as well.
- Also, $p(2) = 2$, since we can put the two elements either in one set together or in separate sets.
- Similarly $p(3) = 5$ several paragraphs above.

- To compute $p(4)$ we use the recurrence relation:

  $P(4) = \sum_{k=0}^{3} C(3,k)p(3-k).$

  $= C(3,0)P(3) + C(3,1)P(2) + C(3,2)p(1) + C(3,3)p(0)$

  $= 1 \cdot 5 + 3 \cdot 2 + 3 \cdot 1 + 1 \cdot 1 = 15$

- Finally, we use the recurrence relation again to find $p(5)$:

  $P(5) = \sum_{k=0}^{4} C(4,k)p(4-k).$

  $=$

  $C(4,0)P(4) + C(4,1)P(3) + C(4,2)p(2) + C(4,3)p(1) + C(4,4)p(0)$

  $= 1 \cdot 15 + 4 \cdot 5 + 6 \cdot 2 + 4 \cdot 1 + 1 \cdot 1 = 52$

- To compute $p(4)$ we use the recurrence relation:

  $P(4) = \sum_{k=0}^{3} C(3,k)p(3-k).$

  $= C(3,0)P(3) + C(3,1)P(2) + C(3,2)p(1) + C(3,3)p(0)$

  $= 1 \cdot 5 + 3 \cdot 2 + 3 \cdot 1 + 1 \cdot 1 = 15$

- Finally, we use the recurrence relation again to find $p(5)$:

  $P(5) = \sum_{k=0}^{4} C(4,k)p(4-k).$

  $=$

  $C(4,0)P(4) + C(4,1)P(3) + C(4,2)p(2) + C(4,3)p(1) + C(4,4)p(0)$

  $= 1 \cdot 15 + 4 \cdot 5 + 6 \cdot 2 + 4 \cdot 1 + 1 \cdot 1 = 52$

- Thus there are exactly 52 partitions of the set (1, 2, 3, 4, 5) (or any other set with five elements).
- It would have been difficult to be sure of obtaining the right answer by trying to list these 52 partitions.

- Thus there are exactly 52 partitions of the set (1, 2, 3, 4, 5) (or any other set with five elements).
- It would have been difficult to be sure of obtaining the right answer by trying to list these 52 partitions.

- The computation of the number of partitions of the natural number n requires one additional wrinkle in the use of recurrence relations.
- Until now we have had only one independent variable, which we usually called n.
- We were finding recurrence relations for functions of one (natural number) variable (i.e., sequences).
- This might be thought of as a one-dimensional problem.

- The computation of the number of partitions of the natural number n requires one additional wrinkle in the use of recurrence relations.
- Until now we have had only one independent variable, which we usually called n.
- We were finding recurrence relations for functions of one (natural number) variable (i.e., sequences).
- This might be thought of as a one-dimensional problem.

- The computation of the number of partitions of the natural number n requires one additional wrinkle in the use of recurrence relations.
- Until now we have had only one independent variable, which we usually called n.
- We were finding recurrence relations for functions of one (natural number) variable (i.e., sequences).
- This might be thought of as a one-dimensional problem.

- The computation of the number of partitions of the natural number n requires one additional wrinkle in the use of recurrence relations.
- Until now we have had only one independent variable, which we usually called n.
- We were finding recurrence relations for functions of one (natural number) variable (i.e., sequences).
- This might be thought of as a one-dimensional problem.

- Recurrence relations also apply to functions of more than one variable.
- In such cases we get recurrence relations for multidimensional arrays, rather than one-dimensional sequences. (It was far from obvious that the trick of introducing an additional variable into the problem of computing the number of partitions of a number was going to be a useful thing to do: think of this idea as the last after a long series of failures by whoever first solved this problem.)

- Recurrence relations also apply to functions of more than one variable.
- In such cases we get recurrence relations for multidimensional arrays, rather than one-dimensional sequences. (It was far from obvious that the trick of introducing an additional variable into the problem of computing the number of partitions of a number was going to be a useful thing to do: think of this idea as the last after a long series of failures by whoever first solved this problem.)

- To compute the number of partitions of the natural number k, which we will denote $P(k)$ as we did previous, we will let $P_m(k)$ denote the number of partitions of the number k into parts no bigger than m each, where m is a positive integer.
- Think of $P_m(k)$ as a function of two variables, k and m.
- For instance, $P(4) = 5$, since the partitions of 4 are 4, 3+1, 2+2, 2+1+1, and 1+1+1+1; but $P_2(4) = 3$, since three of these partitions use only numbers less than or equal to 2.
- We will write down a rather complicated recurrence relation for $p_m(k)$.
- This will solve our problem, since $P(k) = P_k(k)$: No parts bigger than k are used in any partition of k.

- To compute the number of partitions of the natural number k, which we will denote $P(k)$ as we did previous, we will let $P_m(k)$ denote the number of partitions of the number k into parts no bigger than m each, where m is a positive integer.
- Think of $P_m(k)$ as a function of two variables, k and m.
- For instance, $P(4) = 5$, since the partitions of 4 are 4, 3+1, 2+2, 2+1+1, and 1+1+1+1; but $P_2(4) = 3$, since three of these partitions use only numbers less than or equal to 2.
- We will write down a rather complicated recurrence relation for $p_m(k)$.
- This will solve our problem, since $P(k) = P_k(k)$: No parts bigger than k are used in any partition of k.

- To compute the number of partitions of the natural number k, which we will denote $P(k)$ as we did previous, we will let $P_m(k)$ denote the number of partitions of the number k into parts no bigger than m each, where m is a positive integer.
- Think of $P_m(k)$ as a function of two variables, k and m.
- For instance, $P(4) = 5$, since the partitions of 4 are 4, 3+1, 2+2, 2+1+1, and 1+1+1+1; but $P_2(4) = 3$, since three of these partitions use only numbers less than or equal to 2.
- We will write down a rather complicated recurrence relation for $p_m(k)$.
- This will solve our problem, since $P(k) = P_k(k)$: No parts bigger than k are used in any partition of k.

- To compute the number of partitions of the natural number k, which we will denote $P(k)$ as we did previous, we will let $P_m(k)$ denote the number of partitions of the number k into parts no bigger than m each, where m is a positive integer.
- Think of $P_m(k)$ as a function of two variables, k and m.
- For instance, $P(4) = 5$, since the partitions of 4 are 4, 3+1, 2+2, 2+1+1, and 1+1+1+1; but $P_2(4) = 3$, since three of these partitions use only numbers less than or equal to 2.
- We will write down a rather complicated recurrence relation for $p_m(k)$.
- This will solve our problem, since $P(k) = P_k(k)$: No parts bigger than k are used in any partition of k.

- To compute the number of partitions of the natural number k, which we will denote $P(k)$ as we did previous, we will let $P_m(k)$ denote the number of partitions of the number k into parts no bigger than m each, where m is a positive integer.
- Think of $P_m(k)$ as a function of two variables, k and m.
- For instance, $P(4) = 5$, since the partitions of 4 are 4, 3+1, 2+2, 2+1+1, and 1+1+1+1; but $P_2(4) = 3$, since three of these partitions use only numbers less than or equal to 2.
- We will write down a rather complicated recurrence relation for $p_m(k)$.
- This will solve our problem, since $P(k) = P_k(k)$: No parts bigger than k are used in any partition of k.

- As usual, we must ask: How does $P_m(k)$ depend on values of P with smaller arguments? (Both k and m are being viewed as arguments.)
- There are a few special cases, which we can view as the initial conditions for this analysis.
- If $m > k$, then $P_m(k) = P_k(k)$, since no parts bigger than k are ever used in a partition of k: Also, $P_1(k) = 1$, since there is only one way to write k as the sum of numbers no bigger than 1 , namely , $k = 1 + 1 + \cdots + 1$.

- As usual, we must ask: How does $P_m(k)$ depend on values of P with smaller arguments? (Both k and m are being viewed as arguments.)
- There are a few special cases, which we can view as the initial conditions for this analysis.
- If $m > k$, then $P_m(k) = P_k(k)$, since no parts bigger than k are ever used in a partition of k: Also, $P_1(k) = 1$, since there is only one way to write k as the sum of numbers no bigger than 1 , namely , $k = 1 + 1 + \cdots + 1$.

- As usual, we must ask: How does $P_m(k)$ depend on values of P with smaller arguments? (Both k and m are being viewed as arguments.)
- There are a few special cases, which we can view as the initial conditions for this analysis.
- If $m > k$, then $P_m(k) = P_k(k)$, since no parts bigger than k are ever used in a partition of k: Also, $P_1(k) = 1$, since there is only one way to write k as the sum of numbers no bigger than 1 , namely , $k = 1 + 1 + \cdots + 1$.

- Finally , $P_m(0) = 1$, since there is only one partition of 0, namely, the empty sum.
- Now suppose that $1 < m \leqslant k$.
- Then a partition of k into parts no bigger than m is either a partition of k into parts no bigger than $m - 1$ (and there are $P_{m-1}(k)$ of these), or a partition of k into parts no bigger than m, at least one of which is actually m.
- There are $P_m(k - m)$ of the latter type, since once we use one m in the partition, there remains $k - m$ left to be partitioned.

- Finally , $P_m(0) = 1$, since there is only one partition of 0, namely, the empty sum.
- Now suppose that $1 < m \leqslant k$.
- Then a partition of k into parts no bigger than m is either a partition of k into parts no bigger than $m - 1$ (and there are $P_{m-1}(k)$ of these), or a partition of k into parts no bigger than m, at least one of which is actually m.
- There are $P_m(k - m)$ of the latter type, since once we use one m in the partition, there remains $k - m$ left to be partitioned.

- Finally , $P_m(0) = 1$, since there is only one partition of 0, namely, the empty sum.
- Now suppose that $1 < m \leqslant k$.
- Then a partition of k into parts no bigger than m is either a partition of k into parts no bigger than $m - 1$ (and there are $P_{m-1}(k)$ of these), or a partition of k into parts no bigger than m, at least one of which is actually m.
- There are $P_m(k - m)$ of the latter type, since once we use one m in the partition, there remains $k - m$ left to be partitioned.

- Finally , $P_m(0) = 1$, since there is only one partition of 0, namely, the empty sum.
- Now suppose that $1 < m \leqslant k$.
- Then a partition of k into parts no bigger than m is either a partition of k into parts no bigger than $m - 1$ (and there are $P_{m-1}(k)$ of these), or a partition of k into parts no bigger than m, at least one of which is actually m.
- There are $P_m(k - m)$ of the latter type, since once we use one m in the partition, there remains $k - m$ left to be partitioned.

- Therefore, our recurrence relation and initial conditions read as follows:

$$P_m(k) = \begin{cases} 1, & \text{if } k = 0. \\ 1, & \text{if } m = 1. \\ P_k(k), & \text{if } 0 < k < m. \\ P_{m-1}(k) + P_m(k-m), & \text{if } 1 < m \leqslant k. \end{cases} \tag{1}$$

Example

Use the recurrence relation to find the number of partitions of 5.

Solution

- We just apply the recurrence relation repeatedly to compute $P_5(5)$:

$$P_5(5) = P_4(5) + P_5(0) = p_4(5) + 1$$
$$= P_3(5) + P_4(1) + 1 = P_3(5) + P_1(1) + 1 = P_3(5) + 1 + 1 = P_3(5) + 2$$
$$= P_2(5) + P_3(2) + 2 = P_2(5) + P_2(2) + 2 = P_2(5) + P_1(2) + P_2(0) + 2$$
$$= P_2(5) + 4 = P_1(5) + P_2(3) + 4 = P_1(5) + P_1(3) + P_2(1) + 4$$
$$= P_1(5) + P_1(3) + P_1(1) + 4 = 7$$

- Therefore, $P(5) = P_5(5) = 7$ there are seven ways to write 5 as the sum of positive integers, ignoring order.
- The exercises show how recurrence relations can be used to find the number of surjective ("onto") functions, the binomial coefficients, and other similar quantities.

Example

Use the recurrence relation to find the number of partitions of 5.

Solution

- We just apply the recurrence relation repeatedly to compute $P_5(5)$:

$$P_5(5) = P_4(5) + P_5(0) = p_4(5) + 1$$
$$= P_3(5) + P_4(1) + 1 = P_3(5) + P_1(1) + 1 = P_3(5) + 1 + 1 = P_3(5) + 2$$
$$= P_2(5) + P_3(2) + 2 = P_2(5) + P_2(2) + 2 = P_2(5) + P_1(2) + P_2(0) + 2$$
$$= P_2(5) + 4 = P_1(5) + P_2(3) + 4 = P_1(5) + P_1(3) + P_2(1) + 4$$
$$= P_1(5) + P_1(3) + P_1(1) + 4 = 7$$

- Therefore, $P(5) = P_5(5) = 7$ there are seven ways to write 5 as the sum of positive integers, ignoring order.
- The exercises show how recurrence relations can be used to find the number of surjective (”onto”) functions, the binomial coefficients, and other similar quantities.

Example

Use the recurrence relation to find the number of partitions of 5.

Solution

- We just apply the recurrence relation repeatedly to compute $P_5(5)$:

$$P_5(5) = P_4(5) + P_5(0) = p_4(5) + 1$$
$$= P_3(5) + P_4(1) + 1 = P_3(5) + P_1(1) + 1 = P_3(5) + 1 + 1 = P_3(5) + 2$$
$$= P_2(5) + P_3(2) + 2 = P_2(5) + P_2(2) + 2 = P_2(5) + P_1(2) + P_2(0) + 2$$
$$= P_2(5) + 4 = P_1(5) + P_2(3) + 4 = P_1(5) + P_1(3) + P_2(1) + 4$$
$$= P_1(5) + P_1(3) + P_1(1) + 4 = 7$$

- Therefore, $P(5) = P_5(5) = 7$ there are seven ways to write 5 as the sum of positive integers, ignoring order.
- The exercises show how recurrence relations can be used to find the number of surjective ("onto") functions, the binomial coefficients, and other similar quantities.

Example

Use the recurrence relation to find the number of partitions of 5.

Solution

- We just apply the recurrence relation repeatedly to compute $P_5(5)$:

$$
\begin{aligned}
&P_5(5) = P_4(5) + P_5(0) = p_4(5) + 1 \\
&= P_3(5) + P_4(1) + 1 = P_3(5) + P_1(1) + 1 = P_3(5) + 1 + 1 = P_3(5) + 2 \\
&= P_2(5) + P_3(2) + 2 = P_2(5) + P_2(2) + 2 = P_2(5) + P_1(2) + P_2(0) + 2 \\
&= P_2(5) + 4 = P_1(5) + P_2(3) + 4 = P_1(5) + P_1(3) + P_2(1) + 4 \\
&= P_1(5) + P_1(3) + P_1(1) + 4 = 7
\end{aligned}
$$

- Therefore, $P(5) = P_5(5) = 7$ there are seven ways to write 5 as the sum of positive integers, ignoring order.
- The exercises show how recurrence relations can be used to find the number of surjective (”onto”) functions, the binomial coefficients, and other similar quantities.

Example

Use the recurrence relation to find the number of partitions of 5.

Solution

- We just apply the recurrence relation repeatedly to compute $P_5(5)$:

$$
\begin{aligned}
&P_5(5) = P_4(5) + P_5(0) = p_4(5) + 1 \\
&= P_3(5) + P_4(1) + 1 = P_3(5) + P_1(1) + 1 = P_3(5) + 1 + 1 = P_3(5) + 2 \\
&= P_2(5) + P_3(2) + 2 = P_2(5) + P_2(2) + 2 = P_2(5) + P_1(2) + P_2(0) + 2 \\
&= P_2(5) + 4 = P_1(5) + P_2(3) + 4 = P_1(5) + P_1(3) + P_2(1) + 4 \\
&= P_1(5) + P_1(3) + P_1(1) + 4 = 7
\end{aligned}
$$

- Therefore, $P(5) = P_5(5) = 7$ there are seven ways to write 5 as the sum of positive integers, ignoring order.
- The exercises show how recurrence relations can be used to find the number of surjective ("onto") functions, the binomial coefficients, and other similar quantities.

Example

Use the recurrence relation to find the number of partitions of 5.

Solution

- We just apply the recurrence relation repeatedly to compute $P_5(5)$:

$$
\begin{aligned}
&P_5(5) = P_4(5) + P_5(0) = p_4(5) + 1 \\
&= P_3(5) + P_4(1) + 1 = P_3(5) + P_1(1) + 1 = P_3(5) + 1 + 1 = P_3(5) + 2 \\
&= P_2(5) + P_3(2) + 2 = P_2(5) + P_2(2) + 2 = P_2(5) + P_1(2) + P_2(0) + 2 \\
&= P_2(5) + 4 = P_1(5) + P_2(3) + 4 = P_1(5) + P_1(3) + P_2(1) + 4 \\
&= P_1(5) + P_1(3) + P_1(1) + 4 = 7
\end{aligned}
$$

- Therefore, $P(5) = P_5(5) = 7$ there are seven ways to write 5 as the sum of positive integers, ignoring order.
- The exercises show how recurrence relations can be used to find the number of surjective (”onto”) functions, the binomial coefficients, and other similar quantities.

Analysis of Algorithms

- One of the most important applications of recurrence relations is in modeling the performance of recursive algorithms.
- Recall that a recursive algorithm is one that calls itself.
- Let $T(n)$ be the number of steps used by a given recursive algorithm to complete its work on input of size n.
- The algorithm will call itself with an input of size less than n.
- Thus we should be able to write down a recurrence relation expressing $T(n)$ in terms of $T(i)$ for one or more values of i less than n.

Analysis of Algorithms

- One of the most important applications of recurrence relations is in modeling the performance of recursive algorithms.
- Recall that a recursive algorithm is one that calls itself.
- Let $T(n)$ be the number of steps used by a given recursive algorithm to complete its work on input of size n.
- The algorithm will call itself with an input of size less than n.
- Thus we should be able to write down a recurrence relation expressing $T(n)$ in terms of $T(i)$ for one or more values of i less than n.

Analysis of Algorithms

- One of the most important applications of recurrence relations is in modeling the performance of recursive algorithms.
- Recall that a recursive algorithm is one that calls itself.
- Let $T(n)$ be the number of steps used by a given recursive algorithm to complete its work on input of size n.
- The algorithm will call itself with an input of size less than n.
- Thus we should be able to write down a recurrence relation expressing $T(n)$ in terms of $T(i)$ for one or more values of i less than n.

Analysis of Algorithms

- One of the most important applications of recurrence relations is in modeling the performance of recursive algorithms.
- Recall that a recursive algorithm is one that calls itself.
- Let $T(n)$ be the number of steps used by a given recursive algorithm to complete its work on input of size n.
- The algorithm will call itself with an input of size less than n.
- Thus we should be able to write down a recurrence relation expressing $T(n)$ in terms of $T(i)$ for one or more values of i less than n.

Analysis of Algorithms

- One of the most important applications of recurrence relations is in modeling the performance of recursive algorithms.
- Recall that a recursive algorithm is one that calls itself.
- Let $T(n)$ be the number of steps used by a given recursive algorithm to complete its work on input of size n.
- The algorithm will call itself with an input of size less than n.
- Thus we should be able to write down a recurrence relation expressing $T(n)$ in terms of $T(i)$ for one or more values of i less than n.

- We have already seen one example of a recurrence relation used to analyze the efficiency of an algorithm.
- In Section 'Recursive Algorithms', we devised a recursive algorithm to solve the towers of Hanoi problem (in which size is the number of disks).
- We found that we could model the number of disk moves required to transfer n disks with the recurrence relation

$$T(n) = 2T(n-1) + 1, \; n \geqslant 2$$

$$T(1) = 1$$

- We have already seen one example of a recurrence relation used to analyze the efficiency of an algorithm.
- In Section 'Recursive Algorithms', we devised a recursive algorithm to solve the towers of Hanoi problem (in which size is the number of disks).
- We found that we could model the number of disk moves required to transfer n disks with the recurrence relation

$$T(n) = 2T(n-1) + 1, n \geqslant 2$$

$$T(1) = 1$$

- We have already seen one example of a recurrence relation used to analyze the efficiency of an algorithm.
- In Section 'Recursive Algorithms', we devised a recursive algorithm to solve the towers of Hanoi problem (in which size is the number of disks).
- We found that we could model the number of disk moves required to transfer n disks with the recurrence relation

$$T(n) = 2T(n-1) + 1,\ n \geqslant 2$$

$$T(1) = 1$$

- Thus we can compute, for instance, that

$$T(2) = 2T(1) + 1 = 2.1 + 1 = 3$$

$$T(3) = 2T(2) + 1 = 2.3 + 1 = 7$$

$$T(4) = 2T(3) + 1 = 2.7 + 1 = 15$$

and so on.

- Thus we can compute, for instance, that

$$T(2) = 2T(1) + 1 = 2.1 + 1 = 3$$

$$T(3) = 2T(2) + 1 = 2.3 + 1 = 7$$

$$T(4) = 2T(3) + 1 = 2.7 + 1 = 15$$

and so on.

Summary

- **Bell numbers**: The Bell number $p(n)$ is the number of partitions of a set with n elements.
- **Initial conditions for a sequence**: The specification of values for the first few terms of the sequence.
- k^{th}**-order recurrence relation** for the sequence $\{a_n\}$: a recurrence relation that gives an as a function of a_{n-1}, a_{n-2},..., a_{n-k} (e.g., $a_n = 1.04a_{n-1} + a_{n-2}$ is a second order recurrence relation).
- **Recurrence relation** for a sequence: an equation that gives the nth term in a sequence as a function of the previous terms (e.g., $a_n = a_{n-1} + \sqrt{n}$)

Summary

- **Bell numbers**: The Bell number $p(n)$ is the number of partitions of a set with n elements.
- **Initial conditions for a sequence**: The specification of values for the first few terms of the sequence.
- **k^{th}-order recurrence relation** for the sequence $\{a_n\}$: a recurrence relation that gives an as a function of a_{n-1}, a_{n-2},..., a_{n-k} (e.g., $a_n = 1.04a_{n-1} + a_{n-2}$ is a second order recurrence relation).
- **Recurrence relation** for a sequence: an equation that gives the nth term in a sequence as a function of the previous terms (e.g., $a_n = a_{n-1} + \sqrt{n}$)

Summary

- **Bell numbers**: The Bell number $p(n)$ is the number of partitions of a set with n elements.
- **Initial conditions for a sequence**: The specification of values for the first few terms of the sequence.
- k^{th}**-order recurrence relation** for the sequence $\{a_n\}$: a recurrence relation that gives an as a function of a_{n-1}, a_{n-2},..., a_{n-k} (e.g., $a_n = 1.04a_{n-1} + a_{n-2}$ is a second order recurrence relation).
- **Recurrence relation** for a sequence: an equation that gives the nth term in a sequence as a function of the previous terms (e.g., $a_n = a_{n-1} + \sqrt{n}$)

Summary

- **Bell numbers**: The Bell number $p(n)$ is the number of partitions of a set with n elements.
- **Initial conditions for a sequence**: The specification of values for the first few terms of the sequence.
- k^{th}**-order recurrence relation** for the sequence $\{a_n\}$: a recurrence relation that gives an as a function of a_{n-1}, a_{n-2},..., a_{n-k} (e.g., $a_n = 1.04a_{n-1} + a_{n-2}$ is a second order recurrence relation).
- **Recurrence relation** for a sequence: an equation that gives the nth term in a sequence as a function of the previous terms (e.g., $a_n = a_{n-1} + \sqrt{n}$)