Lecture 4

The Cycle Structure of Permutations. Advanced Counting Techniques

October 2015

Permutations and Cycles

Permutations can be thought as rearrangement operations.

Example

1. The permutation $\langle 4, 2, 1, 3\rangle$ maps 1 to 4, 2 to 2, 3 to 1, and 4 to 3. We can write
$$1 \mapsto 4 \mapsto 3 \mapsto 1, \;\; 2 \mapsto 2$$
2. The permutation $\langle 2, 1, 3, 5, 7, 4, 6\rangle$ maps
$$1 \mapsto 2 \mapsto 1, 3 \mapsto 3, \;\; 4 \mapsto 5 \mapsto 7 \mapsto 6 \mapsto 4$$

Definition (Cycle)

A cycle is a map $\pi : \{v_1, v_2, \ldots, v_k\} \to \{v_1, v_2, \ldots, v_k\}$ such that

$$v_1 \mapsto v_2 \mapsto \ldots \mapsto v_{k-1} \mapsto v_k \mapsto v_1$$

The mathematical notation of this cycle is $(v_1, \ldots, v_k)$.
The cycle (v_1) represents the map $\pi : \{v_1\} \to \{v_1\}$ with $\pi(v_1) = v_1$.

The cyclic structure of permutations

Remark

Any permutation can be represented as the composition of disjoint cycles. This kind of representation is called the cyclic structure of a permutation.

Example

1. The permutation $\langle 4, 2, 1, 3\rangle$ can be represented as a composition of 2 disjoint cycles: $(1, 4, 3)(2)$.
2. The permutation $\langle 2, 1, 3, 5, 7, 4, 6\rangle$ can be represented as a composition of 3 disjoint cycles: $(1, 2)(3)(4, 5, 7, 6)$.

The cyclic structure representation of a cycle is not unique: for instance, $(2, 3, 4)$, $(3, 4, 2)$ and $(4, 2, 3)$ are cycles which represent the same function.

The cyclic structure of permutations
Properties

The cyclic structure representation of a cycle is not unique: for instance, $(2, 3, 4)$, $(3, 4, 2)$ and $(4, 2, 3)$ are cycles which represent the same function.

⇒ The cyclic structure of permutations is not unique: for instance, the following cyclic structures represent the same permutation:

- $(1, 5)(2, 3, 4)$
- $(1, 5)(3, 4, 2)$
- $(5, 1)(4, 2, 3)$
- $(2, 3, 4)(1, 5)$
- In general, the cyclic structures produced from each other by
  - rotating the cycles of the structure, to left or right, or
  - permuting the cycles within the cycle structure

  represent the same permutation.

The cyclic structure representation of a cycle is not unique: for instance, $(2,3,4)$, $(3,4,2)$ and $(4,2,3)$ are cycles which represent the same function.

$\Rightarrow$ The cyclic structure of permutations is not unique: for instance, the following cyclic structures represent the same permutation:

- ▷ $(1,5)(2,3,4)$
- ▷ $(1,5)(3,4,2)$
- ▷ $(5,1)(4,2,3)$
- ▷ $(2,3,4)(1,5)$
- ▷ In general, the cyclic structures produced from each other by
    - rotating the cycles of the structure, to left or right, or
    - permuting the cycles within the cycle structure

  represent the same permutation.

- We can define the canonical cyclic structure of a permutation as follows:
    - ▷ Every cycle is written with smallest element first,
    - ▷ Cycles are written in the increasing order of their first element.

Main idea

1. Start computing from 1 the sequence of successors until you reach 1 again. This process builds the first cycle.
2. Choose the smallest element not in the first cycle and build the second cycle in the same manner.
3. Repeat this process until all elements appear in a cycle.

Main idea

1. Start computing from 1 the sequence of successors until you reach 1 again. This process builds the first cycle.
2. Choose the smallest element not in the first cycle and build the second cycle in the same manner.
3. Repeat this process until all elements appear in a cycle.

Exercise

Write down the canonical cyclic structures of the following permutations:

1. $\langle 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \rangle$

Main idea

1. Start computing from 1 the sequence of successors until you reach 1 again. This process builds the first cycle.
2. Choose the smallest element not in the first cycle and build the second cycle in the same manner.
3. Repeat this process until all elements appear in a cycle.

Exercise

Write down the canonical cyclic structures of the following permutations:

1. $\langle 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\rangle$
   $(1)(2)(3)(4)(5)(6)(7)(8)(9)(10)$

Main idea

1. Start computing from 1 the sequence of successors until you reach 1 again. This process builds the first cycle.
2. Choose the smallest element not in the first cycle and build the second cycle in the same manner.
3. Repeat this process until all elements appear in a cycle.

Exercise

Write down the canonical cyclic structures of the following permutations:

1. $\langle 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \rangle$
   $(1)(2)(3)(4)(5)(6)(7)(8)(9)(10)$
2. $\langle 10, 9, 8, 7, 6, 5, 4, 3, 2, 1 \rangle$

Main idea

1. Start computing from 1 the sequence of successors until you reach 1 again. This process builds the first cycle.
2. Choose the smallest element not in the first cycle and build the second cycle in the same manner.
3. Repeat this process until all elements appear in a cycle.

Exercise

Write down the canonical cyclic structures of the following permutations:

1. $\langle 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\rangle$
   $(1)(2)(3)(4)(5)(6)(7)(8)(9)(10)$
2. $\langle 10, 9, 8, 7, 6, 5, 4, 3, 2, 1\rangle$
   $(1, 10)(2, 9)(3, 8)(4, 7)(5, 6)$

Illustrated example

The permutation represented by a cyclic structure $(1,3,4)(2,6,7)(5)$ can be found as follow:

1. Rotate with 1 to the right all cycles of the initial cyclic structure $\Rightarrow (4,1,3)(7,2,6)(5)$
2. Align the cyclic structure produced before on top of the initial cyclic structure:

$$\begin{array}{ccccccccccccccc} (& 4 & , & 1 & , & 3 &)(& 7 & , & 2 & , & 6 &)(& 5 &) \\ & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & \\ (& 1 & , & 3 & , & 4 &)(& 2 & , & 6 & , & 7 &)(& 5 &) \end{array}$$

3. Now, we can read off the corresponding permutation:

$$\begin{array}{ccccccccccccccc} & 1 & & 2 & & 3 & & 4 & & 5 & & 6 & & 7 & \\ & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & \\ \langle & 3 & , & 6 & , & 4 & , & 1 & , & 5 & , & 7 & , & 2 & \rangle \end{array}$$

The type of a permutation π of n elements is the list $\lambda = [\lambda_1, \ldots, \lambda_n]$ where λ_i is the number of cycles of π with length i, for $1 \leq i \leq n$.

The type of a permutation π of n elements is the list $\lambda = [\lambda_1, \ldots, \lambda_n]$ where λ_i is the number of cycles of π with length i, for $1 \leq i \leq n$.

Example

1. $\langle 1,2,3,4,5,6,7\rangle = (1)(2)(3)(4)(5)(6)(7)$ has type $[7,0,0,0,0,0,0]$
2. $\langle 7,6,5,4,3,2,1\rangle = (1,7)(2,6)(3,5)(4)$ has type $[1,3,0,0,0,0,0]$
3. $\langle 1,3,2,6,7,8,9,4,10,5\rangle = (1)(2,3)(4,6,8)(5,7,9,10)$ has type $[1,1,1,1,0,0,0,0,0,0]$

The type of a permutation π of n elements is the list $\lambda = [\lambda_1, \ldots, \lambda_n]$ where λ_i is the number of cycles of π with length i, for $1 \leq i \leq n$.

Example

1. $\langle 1,2,3,4,5,6,7\rangle = (1)(2)(3)(4)(5)(6)(7)$ has type $[7,0,0,0,0,0,0]$
2. $\langle 7,6,5,4,3,2,1\rangle = (1,7)(2,6)(3,5)(4)$ has type $[1,3,0,0,0,0,0]$
3. $\langle 1,3,2,6,7,8,9,4,10,5\rangle = (1)(2,3)(4,6,8)(5,7,9,10)$ has type $[1,1,1,1,0,0,0,0,0,0]$

REMARK: $[\lambda_1, \ldots, \lambda_n]$ is the type of a permutation if and only if $1 \cdot \lambda_1 + 2 \cdot \lambda_2 + \ldots + n \cdot \lambda_n = n$

$i \cdot \lambda_i =$ the number of elements in cycles with length i.

Counting all permutations of a given type

Question: How many permutations have type $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_n]$?

Counting all permutations of a given type

Question: How many permutations have type $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_n]$?

- We write down all $n!$ permutations and insert parentheses in order to build $n!$ cyclic structures of the form

$$\underbrace{c_1^1 \dots c_{\lambda_1}^1}_{\text{cycles with length 1}} \quad \dots \quad \underbrace{c_1^n \dots c_{\lambda_n}^n}_{\text{cycles with length. } n}$$

Counting all permutations of a given type

Question: How many permutations have type $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_n]$?

- We write down all $n!$ permutations and insert parentheses in order to build $n!$ cyclic structures of the form

$$\underbrace{c_1^1 \ldots c_{\lambda_1}^1}_{\text{cycles with length 1}} \quad \ldots \quad \underbrace{c_1^n \ldots c_{\lambda_n}^n}_{\text{cycles with length. } n}$$

- We count the cyclic structures for the same permutation
  - ▷ Every cycle c_k^i of length i can be written in i distinct ways $\Rightarrow$ because of this reason, there are $1^{\lambda_1} \cdot 2^{\lambda_2} \cdot \ldots \cdot n^{\lambda_n}$ cyclic structures which represent the same permutation by Product Rule)
  - ▷ Every permutation of the cycles inside the cyclic structure yields a cyclic structure for the same permutation
    - there are $\lambda_i!$ permutations in every block of cycles of length i

    $\Rightarrow$ for this reason, here are $\lambda_1! \cdot \lambda_2! \cdot \ldots \cdot \lambda_n!$ equivalent cyclic structures. (by Product Rule)

Counting all permutations of a given type

Question: How many permutations have type $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_n]$?

- We write down all $n!$ permutations and insert parentheses in order to build $n!$ cyclic structures of the form

$$\underbrace{c_1^1 \ldots c_{\lambda_1}^1}_{\text{cycles with length 1}} \quad \ldots \quad \underbrace{c_1^n \ldots c_{\lambda_n}^n}_{\text{cycles with length. } n}$$

- We count the cyclic structures for the same permutation
  - ▷ Every cycle c_k^i of length i can be written in i distinct ways $\Rightarrow$ because of this reason, there are $1^{\lambda_1} \cdot 2^{\lambda_2} \cdot \ldots \cdot n^{\lambda_n}$ cyclic structures which represent the same permutation by Product Rule)
  - ▷ Every permutation of the cycles inside the cyclic structure yields a cyclic structure for the same permutation
    - there are $\lambda_i!$ permutations in every block of cycles of length i

    $\Rightarrow$ for this reason, here are $\lambda_1! \cdot \lambda_2! \cdot \ldots \cdot \lambda_n!$ equivalent cyclic structures. (by Product Rule)

$\Rightarrow$ the no. of perms. of type λ is $\dfrac{n!}{\lambda_1! \cdot \lambda_2! \cdot \ldots \cdot \lambda_n! \cdot 1^{\lambda_1} \cdot 2^{\lambda_2} \cdot \ldots \cdot n^{\lambda_n}}$

A useful correspondence

Definition

An integer partition of a positive integer n is a multiset of strictly positive integers whose sum is n.

Note that

Number of integer partition of n = Number of types of n-permutations.

$$[\lambda_1, \ldots, \lambda_n] \leftrightarrow \{\underbrace{1, \ldots, 1}_{\lambda_1 \text{ times}}, \ldots, \underbrace{n, \ldots, n}_{\lambda_n \text{ times}}\}$$

Example

The integer partitions of 5 are the multisets $\{5\}, \{4, 1\}, \{3, 2\}, \{3, 1, 1\}, \{2, 2, 1\}, \{2, 1, 1, 1\}, \{1, 1, 1, 1, 1\}$.
They correspond to the types $[0, 0, 0, 0, 1]$, $[1, 0, 0, 1, 0]$, $[0, 1, 1, 0, 0]$, $[2, 0, 1, 0, 0]$, $[1, 2, 0, 0, 0]$, $[3, 1, 0, 0, 0]$, $[5, 0, 0, 0, 0]$.

- Many interesting counting problems can not be solved with the counting techniques presented so far.
- Examples:
  1. How many *n*-bit strings don't have two consecutive zeroes?
  2. How many ways are there to assign 7 jobs to 3 employees so that each employee is assigned at least one job?

- Many interesting counting problems can not be solved with the counting techniques presented so far.
- Examples:
  1. How many n-bit strings don't have two consecutive zeroes?
  2. How many ways are there to assign 7 jobs to 3 employees so that each employee is assigned at least one job?

  Purpose of this part of the lecture: present more advanced counting techniques:

- Many interesting counting problems can not be solved with the counting techniques presented so far.
- Examples:
  1. How many n-bit strings don't have two consecutive zeroes?
  2. How many ways are there to assign 7 jobs to 3 employees so that each employee is assigned at least one job?

  Purpose of this part of the lecture: present more advanced counting techniques:
  - Recurrence relations

- Many interesting counting problems can not be solved with the counting techniques presented so far.
- Examples:
  1. How many n-bit strings don't have two consecutive zeroes?
  2. How many ways are there to assign 7 jobs to 3 employees so that each employee is assigned at least one job?

  Purpose of this part of the lecture: present more advanced counting techniques:
  - Recurrence relations
  - Solving linear recurrence relations

- Many interesting counting problems can not be solved with the counting techniques presented so far.
- Examples:
    1. How many n-bit strings don't have two consecutive zeroes?
    2. How many ways are there to assign 7 jobs to 3 employees so that each employee is assigned at least one job?

  Purpose of this part of the lecture: present more advanced counting techniques:
    - Recurrence relations
    - Solving linear recurrence relations
    - Divide-and-conquer algorithms

Example

The number of bacteria in a colony doubles every hour. If a colony begins with 5 bacteria, how many will be present in n hours?

ANSWER. Let a_n be the number of bacteria after n hours.

- $a_0 = 5$ (initial knowledge)
- $a_n = 2 \cdot a_{n-1}$ for $n > 0$ (evolution)

Recurrence relations

Example

The number of bacteria in a colony doubles every hour. If a colony begins with 5 bacteria, how many will be present in n hours?

ANSWER. Let a_n be the number of bacteria after n hours.

- $a_0 = 5$ (initial knowledge)
- $a_n = 2 \cdot a_{n-1}$ for $n > 0$ (evolution)

- O recurrence relation for a sequence $\{a_n\}$ is an equation that expresses a_n in terms of one or more of the previous terms $a_0, a_1, \ldots, a_{n-1}$ of the sequence, for all $n \geq n_0$, where $n_0 \geq 0$.

Example

The number of bacteria in a colony doubles every hour. If a colony begins with 5 bacteria, how many will be present in n hours?

ANSWER. Let a_n be the number of bacteria after n hours.

- $a_0 = 5$ (initial knowledge)
- $a_n = 2 \cdot a_{n-1}$ for $n > 0$ (evolution)

- O recurrence relation for a sequence $\{a_n\}$ is an equation that expresses a_n in terms of one or more of the previous terms $a_0, a_1, \ldots, a_{n-1}$ of the sequence, for all $n \geq n_0$, where $n_0 \geq 0$.
- A solution of the recurrence relation is a formula of a_n as a function of n which satisfies the recurrence relation.

Recurrence relations

Example

The number of bacteria in a colony doubles every hour. If a colony begins with 5 bacteria, how many will be present in n hours?

ANSWER. Let a_n be the number of bacteria after n hours.

- $a_0 = 5$ (initial knowledge)
- $a_n = 2 \cdot a_{n-1}$ for $n > 0$ (evolution)

- O recurrence relation for a sequence $\{a_n\}$ is an equation that expresses a_n in terms of one or more of the previous terms $a_0, a_1, \ldots, a_{n-1}$ of the sequence, for all $n \geq n_0$, where $n_0 \geq 0$.
- A solution of the recurrence relation is a formula of a_n as a function of n which satisfies the recurrence relation.

We will develop techniques to solve various kinds of recurrence relations.

- $a_0 = 3, a_1 = 5, a_n = a_{n-1} - a_{n-2}$ for $n \geq 2$.
  All elements of $\{a_n\}$ can be computed recursively:

$$a_2 = a_1 - a_0 = 5 - 3 = 2$$
$$a_3 = a_2 - a_1 = 2 - 5 = -3$$
$$\ldots$$

Can we find a general formula to compute a_n directly, as a function of n?

- $a_0 = 3, a_1 = 5, a_n = a_{n-1} - a_{n-2}$ for $n \geq 2$.
  All elements of $\{a_n\}$ can be computed recursively:

$$a_2 = a_1 - a_0 = 5 - 3 = 2$$
$$a_3 = a_2 - a_1 = 2 - 5 = -3$$
$$\ldots$$

  Can we find a general formula to compute a_n directly, as a function of n?

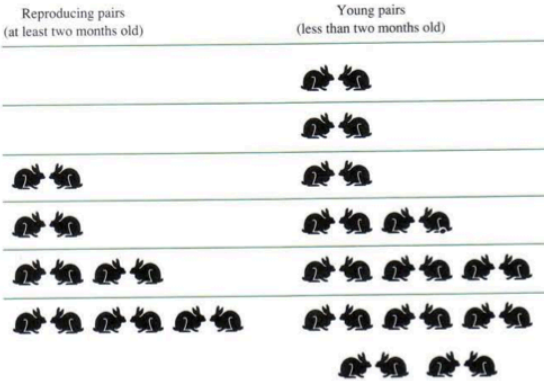
- $a_0 = 0, a_1 = 3, a_n = 2 \cdot a_{n-1} - a_{n-2}$ for $n \geq 2$. All elements of $\{a_n\}$ can be computed recursively:

$$a_2 = 2\,a_1 - a_0 = 6$$
$$a_3 = 2\,a_2 - a_1 = 9$$
$$\ldots$$

  It can be shown by induction on n that $a_n = 3\,n$ for all $n \geq 0$.

Example: Rabbits and Fibonacci numbers

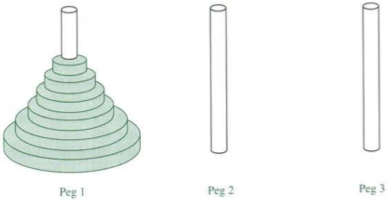
A young pair of rabbits starts breeding when they are 2 months old, by giving birth to another pair each month. Suppose a zero-months old pair of rabbits is placed on an island. Find a recurrence relation for the number of pairs of rabbits on the island after n months.



Month	Reproducing pairs	Young pairs	Total pairs
1	0	1	1
2	0	1	1
3	1	1	2
4	1	2	3
5	2	3	5
6	3	5	8

$f_1 = 1,\ f_2 = 1,\ f_n = f_{n-1} + f_{n-2}$ if $n \geq 2$.

Example: Tower of Hanoi



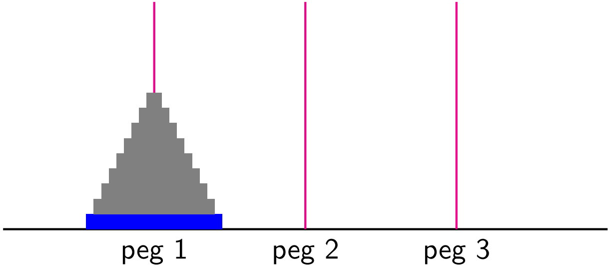
- Move all disks on the second peg in order of size, with the largest disk on the bottom.
- Disks are moved one at a time from one peg to another peg as long as a disk is never placed on top of a smaller disk.

  Question: What is the minimum number of moves needed to solve the Tower of Hanoi problem with n disks?

Example: Tower of Hanoi (continued)

A: Let H_n be the minimum number of moves needed to move n disks in order of size, from one peg to another.

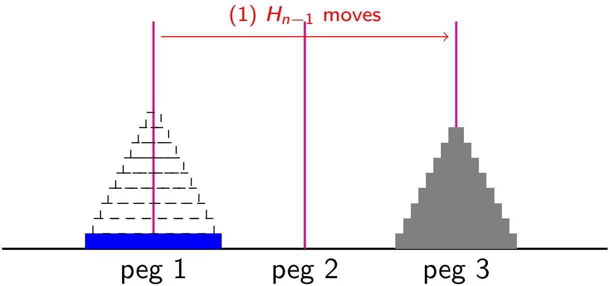
- To place the largest disk on bottom of peg 2, first we must move the $n-1$ smaller disks from peg 1 to peg 3. The minimum number of moves to do so is H_{n-1}.
- After moving the largest disk from peg 1 to peg 2, we need minimum H_{n-1} moves to move the disks from peg 3 to peg 2.



Example: Tower of Hanoi (continued)

A: Let H_n be the minimum number of moves needed to move n disks in order of size, from one peg to another.

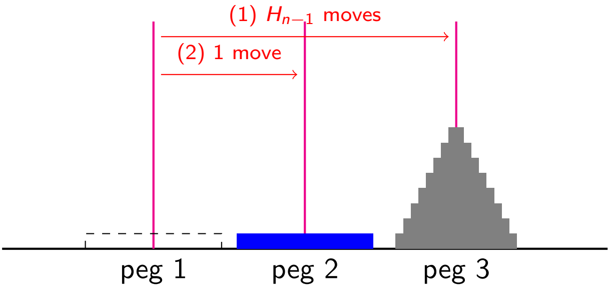
- To place the largest disk on bottom of peg 2, first we must move the $n-1$ smaller disks from peg 1 to peg 3. The minimum number of moves to do so is H_{n-1}.
- After moving the largest disk from peg 1 to peg 2, we need minimum H_{n-1} moves to move the disks from peg 3 to peg 2.



Example: Tower of Hanoi (continued)

A: Let H_n be the minimum number of moves needed to move n disks in order of size, from one peg to another.

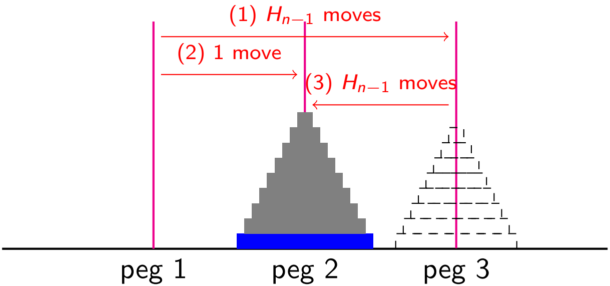
- To place the largest disk on bottom of peg 2, first we must move the $n-1$ smaller disks from peg 1 to peg 3. The minimum number of moves to do so is H_{n-1}.
- After moving the largest disk from peg 1 to peg 2, we need minimum H_{n-1} moves to move the disks from peg 3 to peg 2.



Example: Tower of Hanoi (continued)

A: Let H_n be the minimum number of moves needed to move n disks in order of size, from one peg to another.

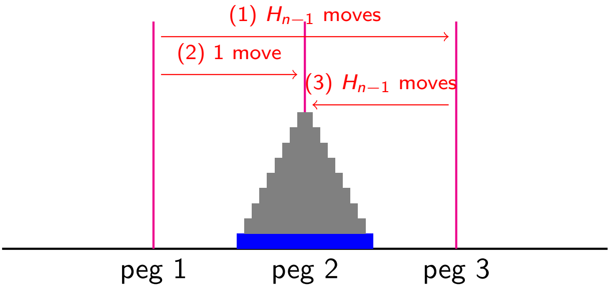
- To place the largest disk on bottom of peg 2, first we must move the $n-1$ smaller disks from peg 1 to peg 3. The minimum number of moves to do so is H_{n-1}.
- After moving the largest disk from peg 1 to peg 2, we need minimum H_{n-1} moves to move the disks from peg 3 to peg 2.



Example: Tower of Hanoi (continued)

A: Let H_n be the minimum number of moves needed to move n disks in order of size, from one peg to another.

- To place the largest disk on bottom of peg 2, first we must move the $n-1$ smaller disks from peg 1 to peg 3. The minimum number of moves to do so is H_{n-1}.
- After moving the largest disk from peg 1 to peg 2, we need minimum H_{n-1} moves to move the disks from peg 3 to peg 2.



$\Rightarrow H_n = H_{n-1} + 1 + H_{n-1} = 2H_{n-1} + 1$. Note that $H_1 = 1$.

Example: Tower of Hanoi (continued)

- We can use an iterative approach to find the formula for H_n when $n > 1$:

$$
\begin{aligned}
H_n &= 2\,H_{n-1} + 1 \\
&= 2(2\,H_{n-2} + 1) + 1 = 2^2\,H_{n-2} + 2 + 1 \\
&= 2^2(2\,H_{n-3} + 1) + 2 + 1 = 2^3\,H_{n-3} + 2^2 + 2 + 1 \\
&\vdots \\
&= 2^{n-1} H_1 + 2^{n-2} + \ldots + 2 + 1 \\
&= 2^{n-1} + 2^{n-2} + \ldots + 2 + 1 \\
&= \frac{2^n - 1}{2 - 1} = 2^n - 1.
\end{aligned}
$$

Example: Tower of Hanoi (continued)

- We can use an iterative approach to find the formula for H_n when $n > 1$:

$$\begin{aligned}
H_n &= 2\,H_{n-1} + 1 \\
&= 2(2\,H_{n-2} + 1) + 1 = 2^2\,H_{n-2} + 2 + 1 \\
&= 2^2(2\,H_{n-3} + 1) + 2 + 1 = 2^3\,H_{n-3} + 2^2 + 2 + 1 \\
&\vdots \\
&= 2^{n-1} H_1 + 2^{n-2} + \ldots + 2 + 1 \\
&= 2^{n-1} + 2^{n-2} + \ldots + 2 + 1 \\
&= \frac{2^n - 1}{2 - 1} = 2^n - 1.
\end{aligned}$$

The myth of the puzzle:

Example: Tower of Hanoi (continued)

- We can use an iterative approach to find the formula for H_n when $n > 1$:

$$\begin{aligned}
H_n &= 2\,H_{n-1} + 1 \\
&= 2(2\,H_{n-2} + 1) + 1 = 2^2\,H_{n-2} + 2 + 1 \\
&= 2^2(2\,H_{n-3} + 1) + 2 + 1 = 2^3\,H_{n-3} + 2^2 + 2 + 1 \\
&\vdots \\
&= 2^{n-1} H_1 + 2^{n-2} + \ldots + 2 + 1 \\
&= 2^{n-1} + 2^{n-2} + \ldots + 2 + 1 \\
&= \frac{2^n - 1}{2 - 1} = 2^n - 1.
\end{aligned}$$

The myth of the puzzle:

- There are 64 disks, and moving 1 disk takes 1 second

Example: Tower of Hanoi (continued)

- We can use an iterative approach to find the formula for H_n when $n > 1$:

$$\begin{aligned} H_n &= 2\,H_{n-1} + 1 \\ &= 2(2\,H_{n-2} + 1) + 1 = 2^2\,H_{n-2} + 2 + 1 \\ &= 2^2(2\,H_{n-3} + 1) + 2 + 1 = 2^3\,H_{n-3} + 2^2 + 2 + 1 \\ &\vdots \\ &= 2^{n-1} H_1 + 2^{n-2} + \ldots + 2 + 1 \\ &= 2^{n-1} + 2^{n-2} + \ldots + 2 + 1 \\ &= \frac{2^n - 1}{2 - 1} = 2^n - 1. \end{aligned}$$

The myth of the puzzle:

- There are 64 disks, and moving 1 disk takes 1 second
- Minimum time to move the Tower of Hanoi= $(2^{64} - 1)\,s = 18446744073709551615\,s \approx 500$ billion years.

Example: Special bit strings

- Find a recurrence relation and give initial conditions for the number of bit strings of length n that do not have two consecutive zeros. How many such bit strings of length 5 do we have?

A: There are 2 disjoint counting tasks:

1. Count the n-bit strings with no 2 consec. 0s that end with 1:
2. Count the n bit-strings with no 2 consec. 0s that end with 0:

Number of bit strings of length n with no two consecutive 0s:

End with a 1: | Any bit string of length $n-1$ with no 2 consecutive 0s | 1 $\qquad a_{n-1}$

End with a 0: | Any bit string of length $n-2$ with no 2 consecutive 0s | 1 0 $\qquad a_{n-2}$

Total: $\qquad a_n = a_{n-1} + a_{n-2}$

The bit strings of length 1 are 0 and 1 $\Rightarrow a_1 = 2$, and the bit strings of length 2 without consecutive 0s are $01, 10, 11 \Rightarrow a_2 = 3$.

Example: Special bit strings (continued)

The number a_n of bit strings of length n without two consecutive zeros is given by the recurrence relation

$$a_1 = 2, \quad a_2 = 3, \quad a_n = a_{n-1} + a_{n-2} \text{ if } n \geq 2.$$

$\Rightarrow$ $a_3 = a_1 + a_2 = 2 + 3 = 5$

$\Rightarrow$ $a_4 = a_2 + a_3 = 3 + 5 = 8$

$\Rightarrow$ $a_5 = a_3 + a_4 = 5 + 8 = 13.$

Linear recurrence relations

- A linear homogeneous recurrence relation of degree k with constant coefficients is a recurrence relation of the form

$$a_n = c_1\, a_{n-1} + c_2\, a_{n-2} + \ldots + c_k\, a_{n-k},$$

where $c_1, c_2, \ldots, c_k \in \mathbb{R}$ and $c_k \neq 0$.

If we know the k initial conditions

- $a_0 = C_0,\ a_1 = C_1,\ \ldots,\ a_{k-1} = C_{k-1},$

then we can compute a_n recursively, for all $n \geq k$.

Example (Linear recurrence relations)

- $\{f_n\}$ where $f_0 = f_1 = 1$, and $f_n = f_{n-1} + f_{n-2}$ if $n > 1$.
- $\{P_n\}$ where $P_0 = 1$, and $P_n = 1.11\, P_{n-1}$ if $n > 0$.

Example (Nonlinear recurrence relations)

$a_0 = 1$, $a_1 = 1$, $a_n = a_{n-1}^2 + a_{n-2}$ for all $n \geq 2$.

Linear recurrence relations

- They occur often in modeling of problems.
- We can find a formula to compute a_n directly form n.

Theorem 1

Consider the recurrence relation

$$a_n = c_1\, a_{n-1} + c_2\, a_{n-2} + \ldots + c_k\, a_{n-k},\;\; a_0 = C_0, \ldots, a_{k-1} = C_{k-1}. \quad (1)$$

Suppose $r_1, \ldots, r_t$ are the distinct roots of $r^k - c_1 r^{k-1} - \ldots - c_k = 0$ with multiplicities $m_1, \ldots, m_t$, respectively, so that $m_i \geq 1$ for $i = 1, 2, \ldots, t$ and $m_1 + m_2 + \ldots + m_t = k$. Then a sequence $\{a_n\}$ is a solution of (1) if and only if

$$\begin{aligned} a_n =& (\alpha_{1,0} + \alpha_{1,1} n + \ldots + \alpha_{1,m_1-1} n^{m_1-1}) r_1^n \\ &+ (\alpha_{2,0} + \alpha_{2,1} n + \ldots + \alpha_{2,m_2-1} n^{m_2-1}) r_2^n \\ &+ \ldots + (\alpha_{t,0} + \alpha_{t,1} n + \ldots + \alpha_{t,m_t-1} n^{m_t-1}) r_t^n \end{aligned}$$

for $n \in \mathbb{N}$, where $\alpha_{i,j}$ are constants for $1 \leq i \leq t$ and $0 \leq j < m_i$.

- Find the solution to the recurrence relation

$$a_n = -3\, a_{n-1} - 3\, a_{n-2} - a_{n-3}$$

with initial conditions $a_0 = 1$, $a_1 = -2$, and $a_2 = -1$.

- Find the solution to the recurrence relation

$$a_n = -3\,a_{n-1} - 3\,a_{n-2} - a_{n-3}$$

with initial conditions $a_0 = 1$, $a_1 = -2$, and $a_2 = -1$.

- ANSWER. The characteristic equation of the recurrence relation is $r^3 + 3r^2 + 3r + 1 = 0$, which has a single root $r = -1$ of multiplicity 3 of the characteristic equation.

- Find the solution to the recurrence relation

$$a_n = -3\,a_{n-1} - 3\,a_{n-2} - a_{n-3}$$

with initial conditions $a_0 = 1$, $a_1 = -2$, and $a_2 = -1$.

- ANSWER. The characteristic equation of the recurrence relation is $r^3 + 3r^2 + 3r + 1 = 0$, which has a single root $r = -1$ of multiplicity 3 of the characteristic equation.
$\Rightarrow$ the solutions of this recurrence relation are of the form

$$a_n = \alpha_{1,0}(-1)^n + \alpha_{1,1}n(-1)^n + \alpha_{1,2}n^2(-1)^n.$$

- Find the solution to the recurrence relation

$$a_n = -3\,a_{n-1} - 3\,a_{n-2} - a_{n-3}$$

with initial conditions $a_0 = 1$, $a_1 = -2$, and $a_2 = -1$.

- ANSWER. The characteristic equation of the recurrence relation is $r^3 + 3r^2 + 3r + 1 = 0$, which has a single root $r = -1$ of multiplicity 3 of the characteristic equation.
$\Rightarrow$ the solutions of this recurrence relation are of the form

$$a_n = \alpha_{1,0}(-1)^n + \alpha_{1,1}n(-1)^n + \alpha_{1,2}n^2(-1)^n.$$

To find the constants $\alpha_{1,0}, \alpha_{1,1}, \alpha_{1,2}$, use the initial conditions:

$$\begin{cases} a_0 = 1 = \alpha_{1,0} \\ a_1 = -2 = -\alpha_{1,0} - \alpha_{1,1} - \alpha_{1,2} \\ a_2 = -1 = \alpha_{1,0} + 2\alpha_{1,1} + 4\alpha_{1,2} \end{cases} \Rightarrow \begin{cases} \alpha_{1,0} = 1 \\ \alpha_{1,1} = 3 \\ \alpha_{1,2} = -2. \end{cases}$$

- Find the solution to the recurrence relation

$$a_n = -3\,a_{n-1} - 3\,a_{n-2} - a_{n-3}$$

with initial conditions $a_0 = 1$, $a_1 = -2$, and $a_2 = -1$.

- ANSWER. The characteristic equation of the recurrence relation is $r^3 + 3r^2 + 3r + 1 = 0$, which has a single root $r = -1$ of multiplicity 3 of the characteristic equation.
$\Rightarrow$ the solutions of this recurrence relation are of the form

$$a_n = \alpha_{1,0}(-1)^n + \alpha_{1,1}n(-1)^n + \alpha_{1,2}n^2(-1)^n.$$

To find the constants $\alpha_{1,0}, \alpha_{1,1}, \alpha_{1,2}$, use the initial conditions:

$$\begin{cases} a_0 = 1 = \alpha_{1,0} \\ a_1 = -2 = -\alpha_{1,0} - \alpha_{1,1} - \alpha_{1,2} \\ a_2 = -1 = \alpha_{1,0} + 2\alpha_{1,1} + 4\alpha_{1,2} \end{cases} \Rightarrow \begin{cases} \alpha_{1,0} = 1 \\ \alpha_{1,1} = 3 \\ \alpha_{1,2} = -2. \end{cases}$$

$\Rightarrow a_n = (1 + 3\,n - 2\,n^2)(-1)^n.$

Definition

A linear non homogeneous recurrence relation with constant coefficients is a recurrence relation of the form

$$a_n = c_1\, a_{n-1} + c_2\, a_{n-2} + \ldots + c_k\, a_{n-k} + F(n),$$

where $c_1, \ldots, c_k \in \mathbb{R}$ and $F(n)$ is a function non identically zero depending only on n. The recurrence relation

$$a_n = c_1\, a_{n-1} + c_2\, a_{n-2} + \ldots + c_k\, a_{n-k}$$

is called the associated recurrence relation.

Nonhomogeneous Recurrences with Constant Coefficients

Definition

A linear non homogeneous recurrence relation with constant coefficients is a recurrence relation of the form

$$a_n = c_1\, a_{n-1} + c_2\, a_{n-2} + \ldots + c_k\, a_{n-k} + F(n),$$

where $c_1, \ldots, c_k \in \mathbb{R}$ and $F(n)$ is a function non identically zero depending only on n. The recurrence relation

$$a_n = c_1\, a_{n-1} + c_2\, a_{n-2} + \ldots + c_k\, a_{n-k}$$

is called the associated recurrence relation.

EXAMPLES

Nonhomogeneous Recurrences with Constant Coefficients

Definition

A linear non homogeneous recurrence relation with constant coefficients is a recurrence relation of the form

$$a_n = c_1\, a_{n-1} + c_2\, a_{n-2} + \ldots + c_k\, a_{n-k} + F(n),$$

where $c_1, \ldots, c_k \in \mathbb{R}$ and $F(n)$ is a function non identically zero depending only on n. The recurrence relation

$$a_n = c_1\, a_{n-1} + c_2\, a_{n-2} + \ldots + c_k\, a_{n-k}$$

is called the associated recurrence relation.

EXAMPLES

1. $a_n = a_{n-1} + 2^n$ is a non homogeneous recurrence relation. The associated homogeneous relation is $a_n = a_{n-1}$.

Nonhomogeneous Recurrences with Constant Coefficients

Definition

A linear non homogeneous recurrence relation with constant coefficients is a recurrence relation of the form

$$a_n = c_1\, a_{n-1} + c_2\, a_{n-2} + \ldots + c_k\, a_{n-k} + F(n),$$

where $c_1, \ldots, c_k \in \mathbb{R}$ and $F(n)$ is a function non identically zero depending only on n. The recurrence relation

$$a_n = c_1\, a_{n-1} + c_2\, a_{n-2} + \ldots + c_k\, a_{n-k}$$

is called the associated recurrence relation.

Examples

1. $a_n = a_{n-1} + 2^n$ is a non homogeneous recurrence relation. The associated homogeneous relation is $a_n = a_{n-1}$.
2. $a_n = a_{n-1} + a_{n-2} + n^2 + n + 1$ is a non homogeneous recurrence relation. The associated homogeneous relation is $a_n = a_{n-1} + a_{n-2}$.

Nonhomogeneous Recurrences with Constant Coefficients

Theorem 2

If $\{a_n^{(p)}\}$ is a particular solution of

$$a_n = c_1\, a_{n-1} + c_2\, a_{n-2} + \ldots + c_k\, a_{n-k} + F(n),$$

then every solution is of the form $\{a_n^{(p)} + a_n^{(h)}\}$, where $\{a_n^{(h)}\}$ is a solution of the associated homogeneous recurrence relation

$$a_n = c_1\, a_{n-1} + c_2\, a_{n-2} + \ldots + c_k\, a_{n-k}.$$

Nonhomogeneous Recurrences with Constant Coefficients

Theorem 2

If $\{a_n^{(p)}\}$ is a particular solution of

$$a_n = c_1\, a_{n-1} + c_2\, a_{n-2} + \ldots + c_k\, a_{n-k} + F(n),$$

then every solution is of the form $\{a_n^{(p)} + a_n^{(h)}\}$, where $\{a_n^{(h)}\}$ is a solution of the associated homogeneous recurrence relation

$$a_n = c_1\, a_{n-1} + c_2\, a_{n-2} + \ldots + c_k\, a_{n-k}.$$

- We know fromTheorem 1 how to compute $\{a_n^{(h)}\}$.

Nonhomogeneous Recurrences with Constant Coefficients

Theorem 2

If $\{a_n^{(p)}\}$ is a particular solution of

$$a_n = c_1\, a_{n-1} + c_2\, a_{n-2} + \ldots + c_k\, a_{n-k} + F(n),$$

then every solution is of the form $\{a_n^{(p)} + a_n^{(h)}\}$, where $\{a_n^{(h)}\}$ is a solution of the associated homogeneous recurrence relation

$$a_n = c_1\, a_{n-1} + c_2\, a_{n-2} + \ldots + c_k\, a_{n-k}.$$

- We know fromTheorem 1 how to compute $\{a_n^{(h)}\}$.

Q: How can we find a particular solution $\{a_n^{(p)}\}$?

Theorem 3

If $F(n) = (b_t\, n^t + b_{t-1}\, n^{t-1} + \ldots + b_1\, t + b_0)\, s^n$ with $b_0, \ldots, b_{t-1}, b_t, s \in \mathbb{R}$ then

1. If s is not a root of the characteristic equation of the associated linear homogeneous recurrence relation, there is a particular solution of the form
$$(p_t\, n^t + p_{t-1}\, n^{t-1} + \ldots + p_1\, n + p_0)\, s^n.$$
2. If s is a root with multiplicity m of the characteristic equation of the associated linear homogeneous recurrence relation, there is a particular solution of the form
$$n^m(p_t n^t + p_{t-1} n^{t-1} + \ldots + p_1\, n + p_0)\, s^n.$$

Q: What is the form of the solution of the nonlinear recursive relation

$$a_n = 6\,a_{n-1} - 9\,a_{n-2} + F(n)$$

when $F(n) = n^2 2^n$?

Q: What is the form of the solution of the nonlinear recursive relation

$$a_n = 6\,a_{n-1} - 9\,a_{n-2} + F(n)$$

when $F(n) = n^2 2^n$?

A: The associated linear homogeneous recurrence relation is $a_n = 6\,a_{n-1} - 9\,a_{n-2}$. It has the characteristic equation $r^2 - 6\,r + 9 = 0$, which has a single root, $r = 3$, of multiplicity 2.

Q: What is the form of the solution of the nonlinear recursive relation

$$a_n = 6\,a_{n-1} - 9\,a_{n-2} + F(n)$$

when $F(n) = n^2 2^n$?

A: The associated linear homogeneous recurrence relation is $a_n = 6\,a_{n-1} - 9\,a_{n-2}$. It has the characteristic equation $r^2 - 6\,r + 9 = 0$, which has a single root, $r = 3$, of multiplicity 2.
$\Rightarrow$ the solution of the homogeneous part is $a_n^{(h)} = (b_1\,n + b_0)\,3^n$.

Q: What is the form of the solution of the nonlinear recursive relation

$$a_n = 6\,a_{n-1} - 9\,a_{n-2} + F(n)$$

when $F(n) = n^2 2^n$?

A: The associated linear homogeneous recurrence relation is $a_n = 6\,a_{n-1} - 9\,a_{n-2}$. It has the characteristic equation $r^2 - 6\,r + 9 = 0$, which has a single root, $r = 3$, of multiplicity 2. $\Rightarrow$ the solution of the homogeneous part is $a_n^{(h)} = (b_1\,n + b_0)\,3^n$.

- $F(n)$ is of the form $Q(n)\,s^n$ where $Q(n)$ is polynomial of degree $t = 2$, and $s = 2$ is not root of the characteristic equation of the associated homogeneous recurrence $\Rightarrow$ by Theorem 3, a particular solution is of the form $a_n^{(p)} = (p_2\,n^2 + p_1\,n + p_0)\,2^n$.

Q: What is the form of the solution of the nonlinear recursive relation

$$a_n = 6\,a_{n-1} - 9\,a_{n-2} + F(n)$$

when $F(n) = n^2 2^n$?

A: The associated linear homogeneous recurrence relation is $a_n = 6\,a_{n-1} - 9\,a_{n-2}$. It has the characteristic equation $r^2 - 6\,r + 9 = 0$, which has a single root, $r = 3$, of multiplicity 2. $\Rightarrow$ the solution of the homogeneous part is $a_n^{(h)} = (b_1\,n + b_0)\,3^n$.

- $F(n)$ is of the form $Q(n)\,s^n$ where $Q(n)$ is polynomial of degree $t = 2$, and $s = 2$ is not root of the characteristic equation of the associated homogeneous recurrence $\Rightarrow$ by Theorem 3, a particular solution is of the form $a_n^{(p)} = (p_2\,n^2 + p_1\,n + p_0)\,2^n$.
- From $a_n^{(p)} = 6\,a_{n-1}^{(p)} - 9\,a_{n-2}^{(p)} + n^2 2^n$ we obtain $2^{n-2}((p_2 - 4)n^2 + (p_1 - 12p_2)n + p_0 - 6\,p_1 + 24\,p_2) = 0$
  $\Rightarrow p_0 = 192, p_1 = 48, p_2 = 4$

Q: What is the form of the solution of the nonlinear recursive relation

$$a_n = 6\,a_{n-1} - 9\,a_{n-2} + F(n)$$

when $F(n) = n^2 2^n$?

A: The associated linear homogeneous recurrence relation is $a_n = 6\,a_{n-1} - 9\,a_{n-2}$. It has the characteristic equation $r^2 - 6\,r + 9 = 0$, which has a single root, $r = 3$, of multiplicity 2.
$\Rightarrow$ the solution of the homogeneous part is $a_n^{(h)} = (b_1\,n + b_0)\,3^n$.

- $F(n)$ is of the form $Q(n)\,s^n$ where $Q(n)$ is polynomial of degree $t = 2$, and $s = 2$ is not root of the characteristic equation of the associated homogeneous recurrence $\Rightarrow$ by Theorem 3, a particular solution is of the form $a_n^{(p)} = (p_2\,n^2 + p_1\,n + p_0)\,2^n$.
- From $a_n^{(p)} = 6\,a_{n-1}^{(p)} - 9\,a_{n-2}^{(p)} + n^2 2^n$ we obtain
$2^{n-2}((p_2 - 4)n^2 + (p_1 - 12p_2)n + p_0 - 6\,p_1 + 24\,p_2) = 0$
$\Rightarrow p_0 = 192, p_1 = 48, p_2 = 4$

$\Rightarrow$ $a_n = a_n^{(p)} + a_n^{(h)} = (4\,n^2 + 48\,n + 192)\,2^n + (b_1\,n + b_0)\,3^n$.

Q: What is the form of the solution of the nonlinear recursive relation $a_n = a_{n-1} + n$ with initial condition $a_1 = 1$?

Q: What is the form of the solution of the nonlinear recursive relation $a_n = a_{n-1} + n$ with initial condition $a_1 = 1$?

A: The associated linear homogeneous recurrence relation for a_n is $a_n = a_{n-1}$. It has the characteristic equation $r - 1 = 0$, thus the solution of it is of the form
$a_n^{(h)} = c \cdot 1^n = c$ where $c \in \mathbb{R}$.

Q: What is the form of the solution of the nonlinear recursive relation $a_n = a_{n-1} + n$ with initial condition $a_1 = 1$?

A: The associated linear homogeneous recurrence relation for a_n is $a_n = a_{n-1}$. It has the characteristic equation $r - 1 = 0$, thus the solution of it is of the form
$a_n^{(h)} = c \cdot 1^n = c$ where $c \in \mathbb{R}$.

- The nonlinear part is $F(n) = Q(n)\, s^n$ where $Q(n) = n$ and $s = 1$ is a solution with multiplicity 1 of the characteristic equation of the associated linear homogeneous recurrence relation $\Rightarrow$ by Theorem 3, a particular solution is of the form
$a_n^{(p)} = n^s(p_1\, n + p_0)\, 1^n = p_1 n^2 + p_0 n$.

Q: What is the form of the solution of the nonlinear recursive relation $a_n = a_{n-1} + n$ with initial condition $a_1 = 1$?

A: The associated linear homogeneous recurrence relation for a_n is $a_n = a_{n-1}$. It has the characteristic equation $r - 1 = 0$, thus the solution of it is of the form
$a_n^{(h)} = c \cdot 1^n = c$ where $c \in \mathbb{R}$.

- The nonlinear part is $F(n) = Q(n)\, s^n$ where $Q(n) = n$ and $s = 1$ is a solution with multiplicity 1 of the characteristic equation of the associated linear homogeneous recurrence relation $\Rightarrow$ by Theorem 3, a particular solution is of the form
$a_n^{(p)} = n^s(p_1\, n + p_0)\, 1^n = p_1 n^2 + p_0 n$.
- To find the values of p_0 and p_1, we know that $a_n^{(p)} = a_{n-1}^{(p)} + n$, which implies $n(2\, p_1 - 1) + (p_0 - p_1) = 0$, which means that $p_0 = p_1 = \frac{1}{2}$. Hence $a_n^{(p)} = \frac{n^2}{2} + \frac{n}{2} = \frac{n(n+1)}{2}$.

Q: What is the form of the solution of the nonlinear recursive relation $a_n = a_{n-1} + n$ with initial condition $a_1 = 1$?

A: The associated linear homogeneous recurrence relation for a_n is $a_n = a_{n-1}$. It has the characteristic equation $r - 1 = 0$, thus the solution of it is of the form
$a_n^{(h)} = c \cdot 1^n = c$ where $c \in \mathbb{R}$.

- The nonlinear part is $F(n) = Q(n)\, s^n$ where $Q(n) = n$ and $s = 1$ is a solution with multiplicity 1 of the characteristic equation of the associated linear homogeneous recurrence relation $\Rightarrow$ by Theorem 3, a particular solution is of the form
$a_n^{(p)} = n^s(p_1\, n + p_0)\, 1^n = p_1 n^2 + p_0 n$.
- To find the values of p_0 and p_1, we know that $a_n^{(p)} = a_{n-1}^{(p)} + n$, which implies $n(2\, p_1 - 1) + (p_0 - p_1) = 0$, which means that $p_0 = p_1 = \frac{1}{2}$. Hence $a_n^{(p)} = \frac{n^2}{2} + \frac{n}{2} = \frac{n(n+1)}{2}$.
- By Theorem 2, we have $a_n = a_n^{(p)} + a_n^{(h)} = c + \frac{n(n+1)}{2}$. Also, we have $1 = a_1 = c + \frac{1 \cdot 2}{2} = c + 1$, so $c = 0$. Thus $a_n = \frac{n(n+1)}{2}$.

How do they work?

Divide a problem into one or more instances of the same problem, but of smaller size.

Conquer the problem by using the solutions of the smaller problems to find a solution of the original problem.

Divide-and-Conquer algorithms and recurrences

How do they work?

Divide a problem into one or more instances of the same problem, but of smaller size.

Conquer the problem by using the solutions of the smaller problems to find a solution of the original problem.

Typical examples:

1. Binary search for an element in a sorted list.
2. Sorting a list by successively splitting the list into halves, and sort each half separately.
3. ...

Divide-and-Conquer recurrence relations

Phases of a divide-and-conquer algorithm

- Divide a problem of size n into b subproblems of size n/b.
  - REMARK. In reality, not all subproblems have exactly the same size: some have size $\lceil n/b \rceil$, other have size $\lfloor n/b \rfloor$.
- ASSUMPTIONS
  - $f(n/b) :=$ number of operations required to solve problems of size n/b
  - $a :=$ number of subproblems that that have to be solved.
  - $g(n) :=$ number of extra operations required to combine the solutions of subproblems into a solution of the initial problem (the conquer step)

$\Rightarrow$ $f(n) = a\, f(n/b) + g(n).$

This is called a divide-and-conquer recurrence relation.

Search an item in a sorted sequence of n items, as follows:

- Split the initial sorted sequence into 2 sorted sequences of size $n/2$, and choose the subsequence in which to search further $\Rightarrow$ one subproblem of size $n/2$,
- 2 comparisons are needed to determine:
  1. which half of the sequence to use, and
  2. if there are any elements in the list.
- $\Rightarrow$ divide-and-conquer relation

$$f(n) = f(n/2) + 2.$$

procedure MergeSort($L = a_1, \ldots, a_n$)
if $n > 1$ **then**
$m = \lfloor n/2 \rfloor$
$L_1 = a_1, \ldots, a_m$
$L_2 = a_{m+1}, \ldots, a_n$
$L := \texttt{merge}(\textsc{MergeSort}(L1), \textsc{MergeSort}(L2))$
/* L is now sorted into elements in nondecreasing order */

procedure merge(L_1, L_2: sorted list)
L :=empty list
while L_1 and L_2 are both non-empty
remove smaller of first element of L_1 and L_2 from the list it is in, and put it at the right end of L.
if removal of this element makes one list empty
then remove all elements from the other list and append them to L.

Merging the sorted lists 2,3,5,6 and 1,4.			
First list	Second list	Merged list	Comparison
2,3,5,6	1,4		$1 < 2$
2,3,5,6	4	1	$2 < 4$
3,5,6	4	1,2	$3 < 4$
5,6	4	1,2,3	$4 < 5$
5,6		1,2,3,4	
		1,2,3,4,5,6	

Merging the sorted lists 2,3,5,6 and 1,4.			
First list	Second list	Merged list	Comparison
2,3,5,6	1,4		$1 < 2$
2,3,5,6	4	1	$2 < 4$
3,5,6	4	1,2	$3 < 4$
5,6	4	1,2,3	$4 < 5$
5,6		1,2,3,4	
		1,2,3,4,5,6	

Remarks

1. MergeSort uses fewer than n comparisons to merge 2 lists with $n/2$ elements each.
2. The number of comparisons used by MergeSort to sort a list of n elements is less than $M(n)$, where

$$M(n) = 2\,M(n/2) + n.$$

Theorem 4

Let f be an increasing function that satisfies the recurrence relation

$$f(n) = a\,f(n/b) + c$$

whenever n is divisible by b, where $a \geq 1$, b is an integer greater than 1, and $c \in \mathbb{R}$ is positive. Then

$$f(n) \text{ is } \begin{cases} O(n^{\log_b(a)}) & \text{if } a > 1 \\ O(\log n) & \text{if } a = 1. \end{cases}$$

Furthermore, when $n = b^k$, where k is a positive integer, then

$$f(n) = C_1\, n^{\log_b a} + C_2,$$

where $C_1 = f(1) + C/(a-1)$ and $C_2 = -c/(a-1)$.

Theorem 5 (Master Theorem)

Let f be an increasing function that satisfies the recurrence relation

$$f(n) = a\, f(n/b) + c\, n^d$$

whenever $n = b^k$, where k is a positive integer, $a \geq 1$, b is an integer greater than 1, and $c, d \in \mathbb{R}$ with $c > 0$ and $d \geq 0$. Then

$$f(n) \text{ is } \begin{cases} O(n^d) & \text{if } a < b^d, \\ O(n^d \log n) & \text{if } a = b^d, \\ O(n^{\log_b a}) & \text{if } a > b^d. \end{cases}$$

Theorem 5 (Master Theorem)

Let f be an increasing function that satisfies the recurrence relation

$$f(n) = a\,f(n/b) + c\,n^d$$

whenever $n = b^k$, where k is a positive integer, $a \geq 1$, b is an integer greater than 1, and $c, d \in \mathbb{R}$ with $c > 0$ and $d \geq 0$. Then

$$f(n) \text{ is } \begin{cases} O(n^d) & \text{if } a < b^d, \\ O(n^d \log n) & \text{if } a = b^d, \\ O(n^{\log_b a}) & \text{if } a > b^d. \end{cases}$$

Example (Complexity of MergeSort)

$M(n) = a\,M(n/b) + c\,n^d$ where $a = b = 2$, $c = d = 1$
$\Rightarrow M(n)$ is $O(n \log n)$.

References

Section 3.1 from

- S. Pemmaraju, S. Skiena. *Combinatorics and Graph Theory with Mathematica*. Cambridge University Press. 2003.

Chapter 7 of

- Kenneth H. Rosen. *Discrete Mathematics and Its Applications.* Sixth Edition. McGraw-Hill, 2007.