Introduction to Discrete Math

Felipe P. Vista IV

Course Outline

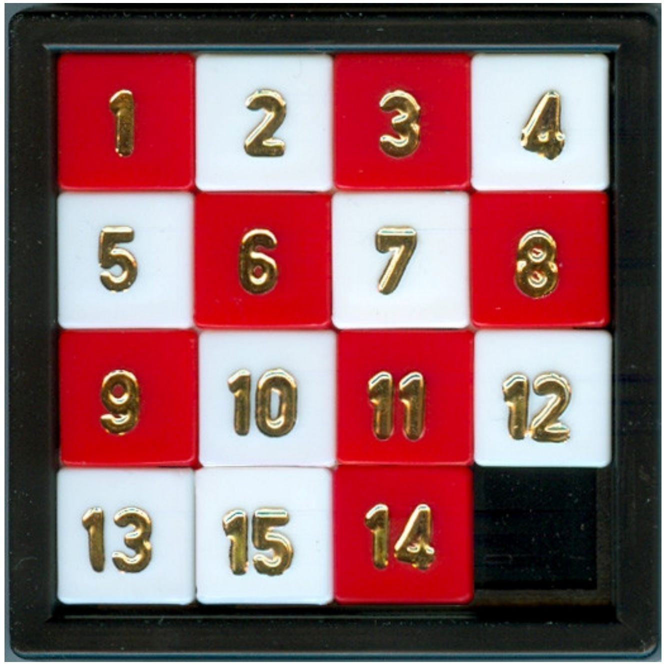
- Mathematical Thinking
  - Convincing Arguments, Find Example, Recursion, Logic, Invariants
- Probability & Combinatronics
  - Counting, Probability, Random Variables
- Graph Theory
  - Graphs (cycles, classes, parameters)
- Number Theory & Cryptography
  - Arithmetic in modular form
  - Intro to Cryptography

Mathematical Thinking – Invariants

THE 15-PUZZLE

- The game
- Permutations
- Proof: The Challenging Part
- Mission Impossible
- Classify a Permutation

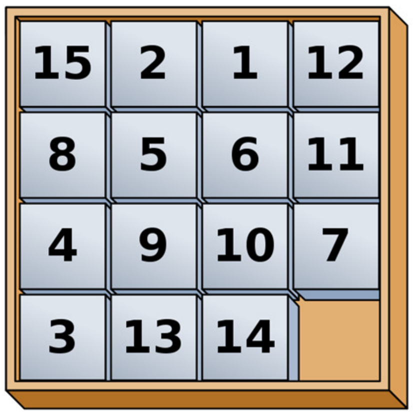
The 15-Puzzle



https://upload.wikimedia.org/wikipedia/commons/4/48/15-Puzzle.jpg

The Game

- move the pieces (into an empty neighbor square)
- goal → to obtain a particular configuration
- go back to starting configuration



https://upload.wikimedia.org/wikipedia/commons/thumb/f/ff/15-puzzle_magical.svg/800px-15-puzzle_magical.svg.png

https://i.stack.imgur.com/0B14h.png

The Game

- move the pieces (into an empty neighbor square)

The Game

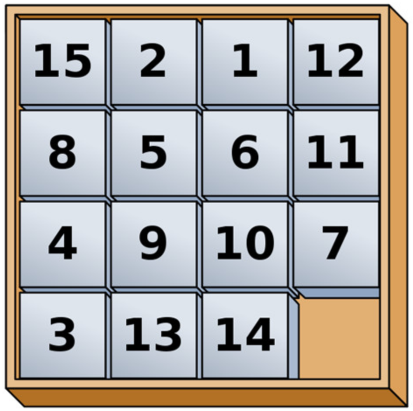
- move the pieces (into an empty neighbor square)
- goal → to obtain a particular configuration

The Game

- move the pieces (into an empty neighbor square)
- goal → to obtain a particular configuration
- go back to starting configuration

The Game

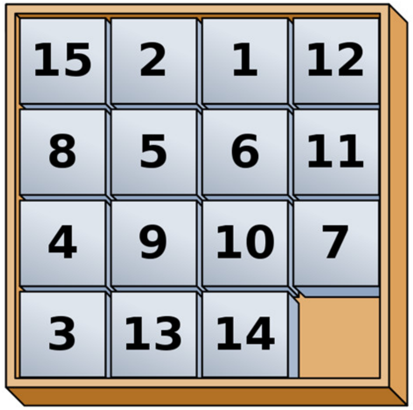
- move the pieces (into an empty neighbor square)
- goal → to obtain a particular configuration
- go back to starting configuration



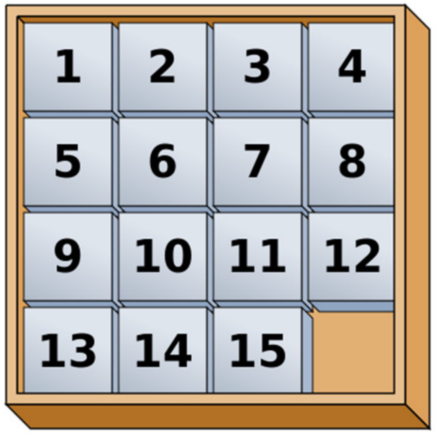
https://upload.wikimedia.org/wikipedia/commons/thumb/f/ff
/15-puzzle_magical.svg/800px-15-puzzle_magical.svg.png

The Game

- move the pieces (into an empty neighbor square)
- goal → to obtain a particular configuration
- go back to starting configuration



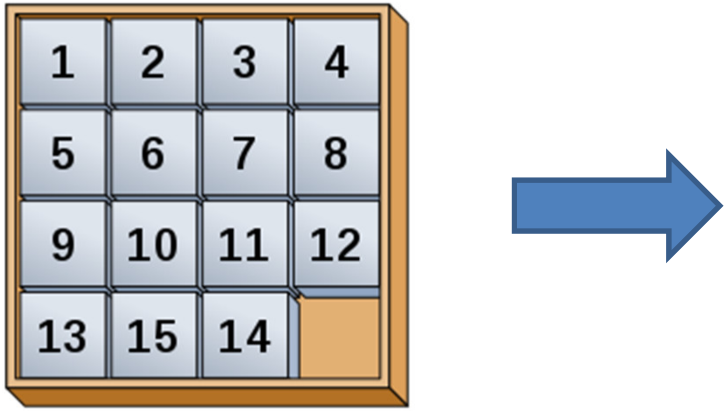
https://upload.wikimedia.org/wikipedia/commons/thumb/f/ff/15-puzzle_magical.svg/800px-15-puzzle_magical.svg.png



https://i.stack.imgur.com/0B14h.png

The Game

- move the pieces (into an empty neighbor square)
- goal → to obtain a particular configuration
- go back to starting configuration
- Are you up to it?



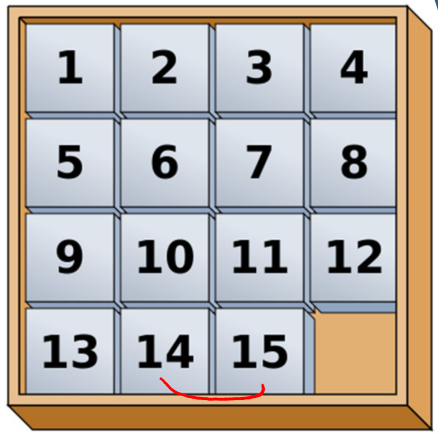
https://upload.wikimedia.org/wikipedia/commons/thumb/3/39/15-puzzle-loyd.svg/600px-15-puzzle-loyd.svg.png

The Game

- move the pieces (into an empty neighbor square)
- goal → to obtain a particular configuration
- go back to starting configuration
- Are you up to it?

$100 Dare!

https://upload.wikimedia.org/wikipedia/commons/thumb/3/39/15-puzzle-loyd.svg/600px-15-puzzle-loyd.svg.png



https://i.stack.imgur.com/0B14h.png

History

- invented in the 19^{th} century

History

- invented in the 19th century
- complicated claims (Chapman, Sam Loyd)

History

- invented in the 19th century
- complicated claims (Chapman, Sam Loyd)
- very popular for some time, prizes were offered!

History

- invented in the 19th century
- complicated claims (Chapman, Sam Loyd)
- very popular for some time, prizes were offered!
- why take risk of giving prize (15-14 interchanged):

History

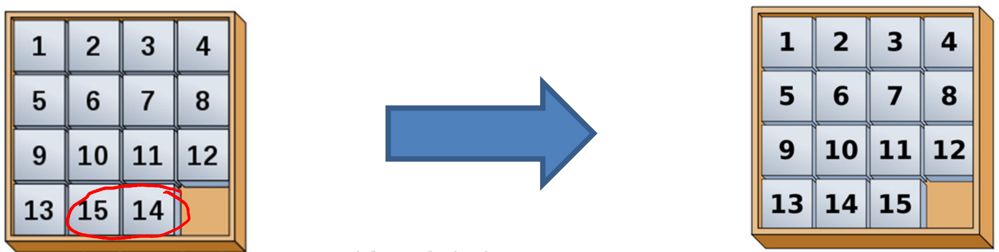
- invented in the 19th century
- complicated claims (Chapman, Sam Loyd)
- very popular for some time, prizes were offered!
- why take risk of giving prize (15-14 interchanged):
  - impossibility proof exists!

History

- invented in the 19th century
- complicated claims (Chapman, Sam Loyd)
- very popular for some time, prizes were offered!
- why take risk of giving prize (15-14 interchanged):
  - impossibility proof exists!
- Challenge: reinvent this proof!

History

- invented in the 19th century
- complicated claims (Chapman, Sam Loyd)
- very popular for some time, prizes were offered!
- why take risk of giving prize (15-14 interchanged):
  - impossibility proof exists!
- Challenge: reinvent this proof!



https://upload.wikimedia.org/wikipedia/commons/thumb/3/39/15-puzzle-loyd.svg/600px-15-puzzle-loyd.svg.png

https://i.stack.imgur.com/0B14h.png

- The game
- Permutations
- Proof: The Challenging Part
- Mission Impossible
- Classify a Permutation

Another point of view

- Empty cell active

Another point of view

- Empty cell active
  - move around, exchange places with neighbors

Another point of view

- Empty cell active
  - move around, exchange places with neighbors
- Generally:

Another point of view

- Empty cell active
  - move around, exchange places with neighbors
- Generally:
  - permutations of n objects obtained through sequence of pair exchanges (transpositions)

Another point of view

- Empty cell active
  - move around, exchange places with neighbors
- Generally:
  - permutations of n objects obtained through sequence of pair exchanges (transpositions)
  - STOP → SPOT:

Another point of view

- Empty cell active
  - move around, exchange places with neighbors
- Generally:
  - permutations of n objects obtained through sequence of pair exchanges (transpositions)
  - STOP → SPOT:
    - one transposition enough

Another point of view

- Empty cell active
  - move around, exchange places with neighbors
- Generally:
  - permutations of n objects obtained through sequence of pair exchanges (transpositions)
  - STOP → SPOT:
    - one transposition enough
  - STOP → POST:

STOP → SPOT

STOP → POST

steps?

Another point of view

- Empty cell active
  - move around, exchange places with neighbors
- Generally:
  - permutations of n objects obtained through sequence of pair exchanges (transpositions)
  - STOP → SPOT:
    - one transposition enough
  - STOP →POST:
    - how many transposition?

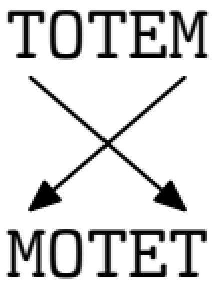
STOP → POST
PTOS
POTS
POST

Even and Odd Permutations

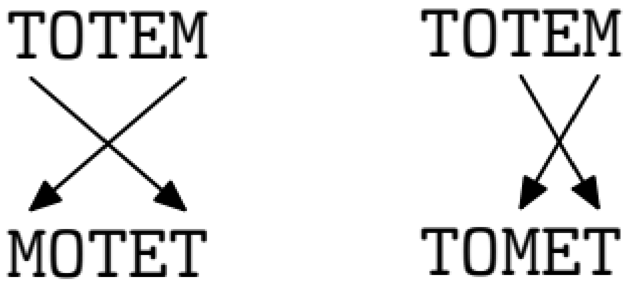
- STOP $\to$ SPOT: 1, 3, 5, 7, …
- STOP $\to$ POST: 3, 5, 7, …
- STOP $\to$ POTS: 2, 4, 6, …
- $n \to n+2$ transposition: twice nothing
- Conjecture: permutations are two types
  - Even
  - Odd

A Counterexample

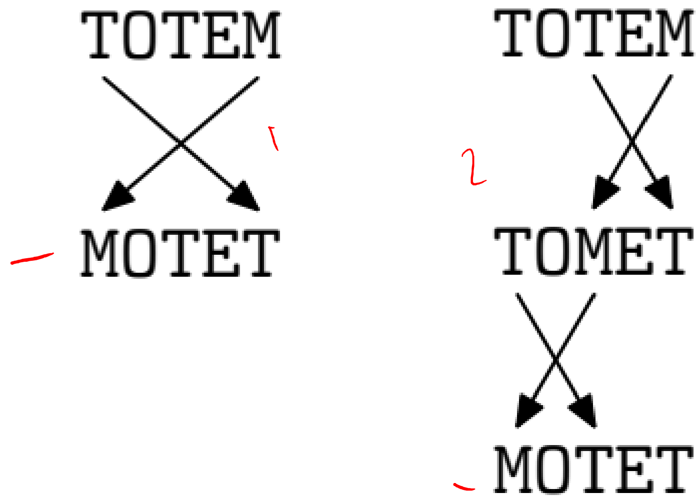
A Counterexample



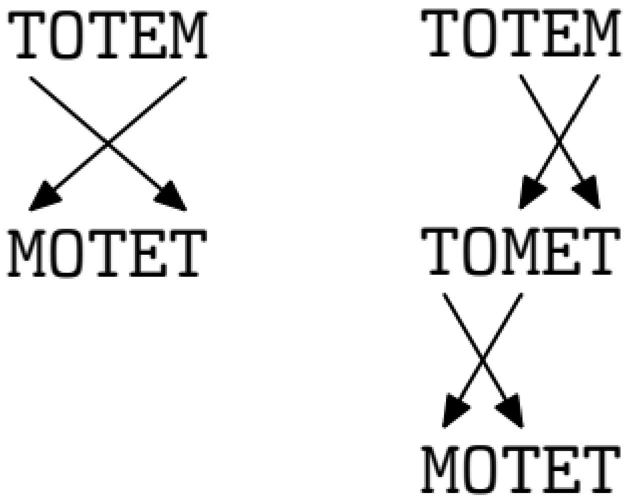
A Counterexample



A Counterexample

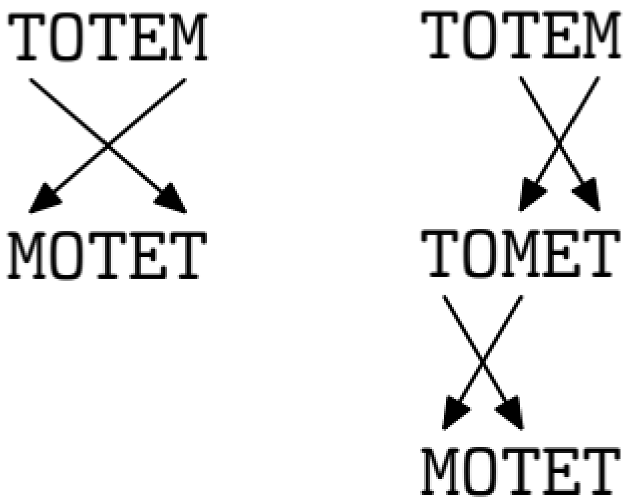


A Counterexample



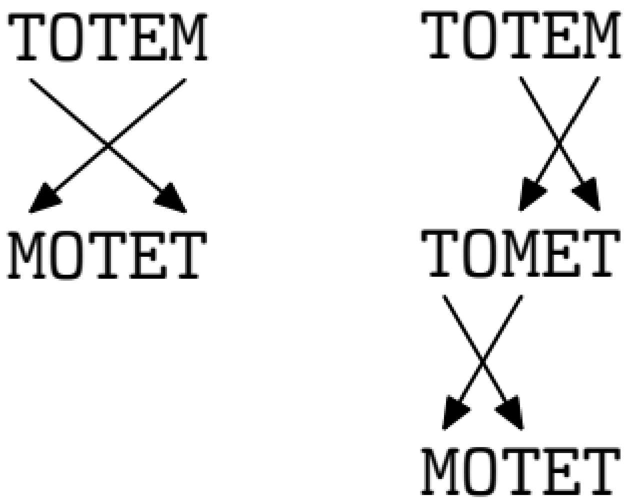
- even and odd at the same time?

A Counterexample



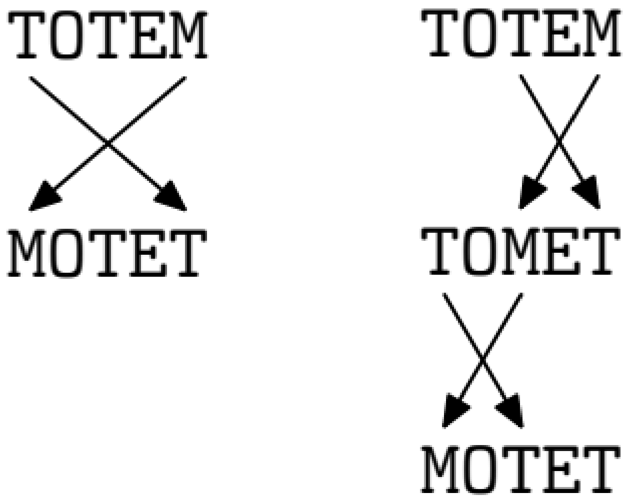
- even and odd at the same time?
- spoiler!

A Counterexample



- even and odd at the same time?
- spoiler!
  - two T's are mixed (we assumed all letters are different)

A Counterexample



- even and odd at the same time?
- spoiler!
  - two T's are mixed (we assumed all letters are different)

Theorem

Theorem

STOP → STPO
SOTP

- each permutation can be obtained through transpositions

Theorem

- each permutation can be obtained through transpositions
- some permutations can be derived only through an even number of transpositions, while others can be derived only through odd number of transpositions

Proof: The Easy Part

Proof: The Easy Part

- claim: each permutation can be obtained by transpositions

Proof: The Easy Part

- claim: each permutation can be obtained by transpositions
- proof: put letters to their right place through one transposition per letter

Proof: The Easy Part

- claim: each permutation can be obtained by transpositions
- proof: put letters to their right place through one transposition per letter

$$\text{STOP} \rightarrow \text{POST}$$

Proof: The Easy Part

- claim: each permutation can be obtained by transpositions
- proof: put letters to their right place through one transposition per letter

STOP → POST
STOP

Proof: The Easy Part

- claim: each permutation can be obtained by transpositions
- proof: put letters to their right place through one transposition per letter

STOP → POST

STOP

Proof: The Easy Part

- claim: each permutation can be obtained by transpositions
- proof: put letters to their right place through one transposition per letter

STOP → POST

STOP

PTOS

Proof: The Easy Part

- claim: each permutation can be obtained by transpositions
- proof: put letters to their right place through one transposition per letter

$$\text{STOP} \rightarrow \text{POST}$$
$$\text{STOP}$$
$$\text{PTOS}$$

Proof: The Easy Part

- claim: each permutation can be obtained by transpositions
- proof: put letters to their right place through one transposition per letter

STOP → POST

STOP

PTOS

POTS

Proof: The Easy Part

- claim: each permutation can be obtained by transpositions
- proof: put letters to their right place through one transposition per letter

STOP → POST

STOP

PTOS

POTS

Proof: The Easy Part

- claim: each permutation can be obtained by transpositions
- proof: put letters to their right place through one transposition per letter

permutation

STOP → POST

STOP
PTOS
POTS
POST

Reminder

- Everybody, make sure that your name in ZOOM is in the following format:
  - Ex: 202054321 Juan Dela Cruz

Not changing your name to this format

* you **will** be marked Absent * → absent?

- The game
- Permutations
- Proof: The Challenging Part
- Mission Impossible
- Classify a Permutation

Even is Not Odd

Even is Not Odd

- second claim: the same permutation cannot be even and odd at the same time

Even is Not Odd

- second claim: the same permutation cannot be even and odd at the same time
- special case: *an identity permutation* (no changes) *cannot be derived through odd number of transpositions*

Even is Not Odd

- second claim: the same permutation cannot be even and odd at the same time
- special case: *an identity permutation* (no changes) *cannot be derived through odd number of transpositions*
- general case can be reduced to this special case:

Even is Not Odd

- second claim: the same permutation cannot be even and odd at the same time
- special case: *an identity permutation* (no changes) *cannot be derived through odd number of transpositions*
- general case can be reduced to this special case:

$$A \to \ldots \to B \text{ (even)}, A \to \ldots \to B \text{ (odd)}$$

Even is Not Odd

- second claim: the same permutation cannot be even and odd at the same time
- special case: *an identity permutation* (no changes) *cannot be derived through odd number of transpositions*
- general case can be reduced to this special case:

$$A \to \ldots \to B \text{ (even)}, A \to \ldots \to B \text{ (odd)}$$

  – back and forth: $even + odd = odd$ number of transpositions brings us back

Even is Not Odd

- second claim: the same permutation cannot be even and odd at the same time
- special case: *an identity permutation* (no changes) *cannot be derived through odd number of transpositions*
- general case can be reduced to this special case:

  $$A \to \ldots \to B \text{ (even)}, A \to \ldots \to B \text{ (odd)}$$

  – back and forth: *even* + *odd* = *odd* number of transpositions brings us back

  – Not possible if we believe in the special case

A B C D A C B D A B D C A C B D

odd 1 even 2

Transposition of Neighbors

Transposition of Neighbors

- claim: after odd number of neighbor transpositions, we cannot return to original position

Transposition of Neighbors

- claim: after odd number of neighbor transpositions, we cannot return to original position
- look at every pair of letters, why is the number of transpositions for this pair even?

Transposition of Neighbors

- claim: after odd number of neighbor transpositions, we cannot return to original position
- look at every pair of letters, why is the number of transpositions for this pair even?
  - Each transposition of the pair changes the order (who is on the left)

Transposition of Neighbors

- claim: after odd number of neighbor transpositions, we cannot return to original position
- look at every pair of letters, why is the number of transpositions for this pair even?
  - Each transposition of the pair changes the order (who is on the left)
- note that neighbor transposition does not change order in other pairs

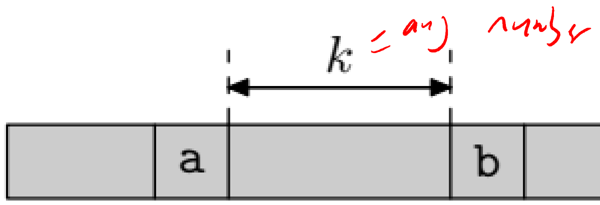
a b = 2k +1 → odd

b a = 2k +1 → odd

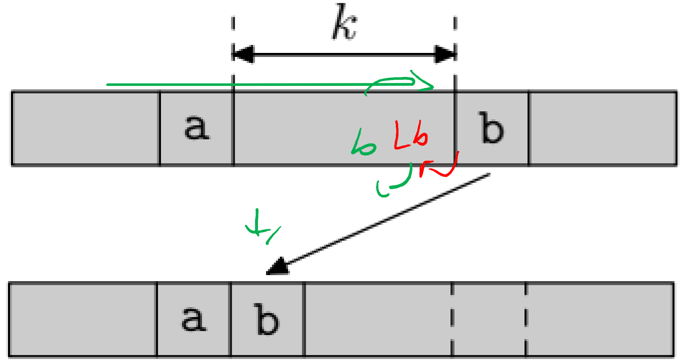
even

Reduction to the Neighbor Case

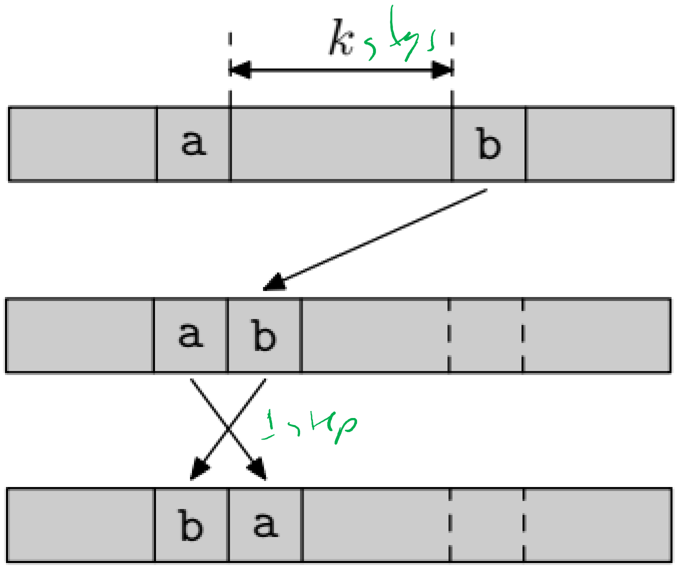
Reduction to the Neighbor Case



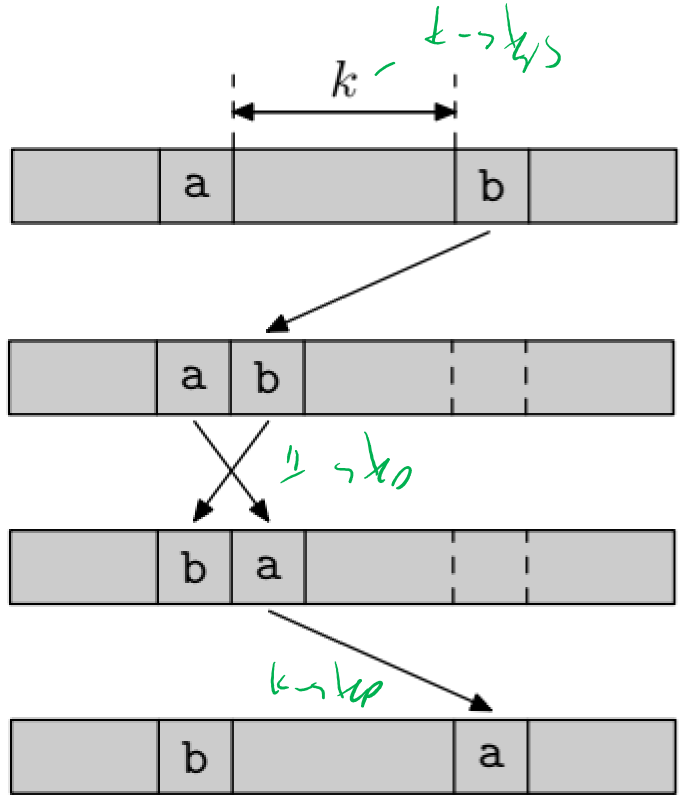
Reduction to the Neighbor Case

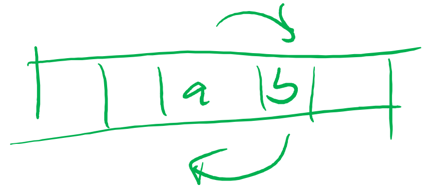


Reduction to the Neighbor Case

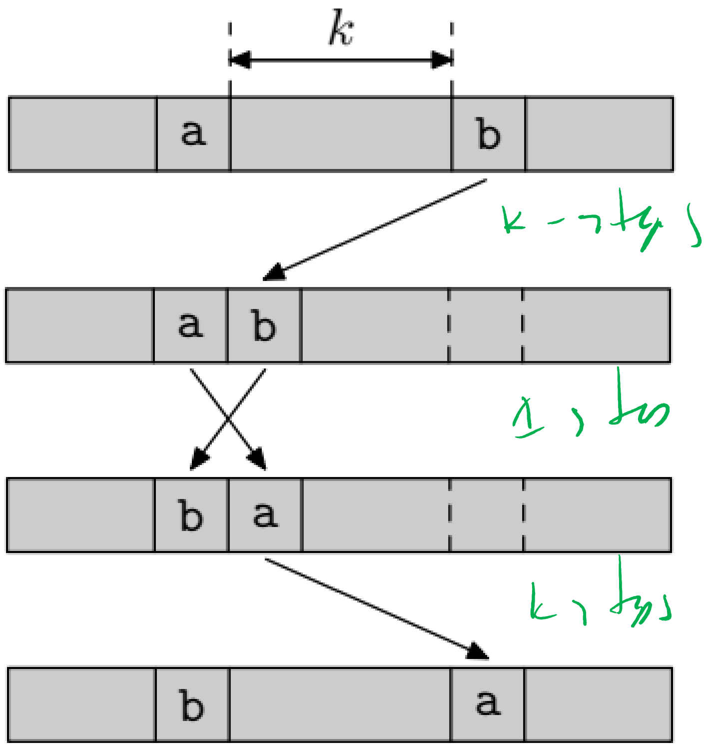


Reduction to the Neighbor Case



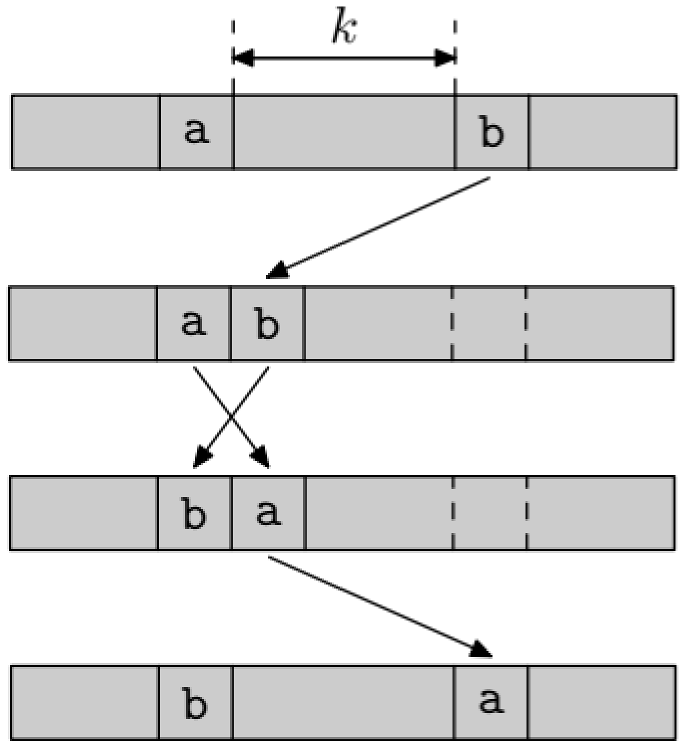


Reduction to the Neighbor Case



$k + 1 + k = 2k + 1$ neighbor transpositions

Reduction to the Neighbor Case



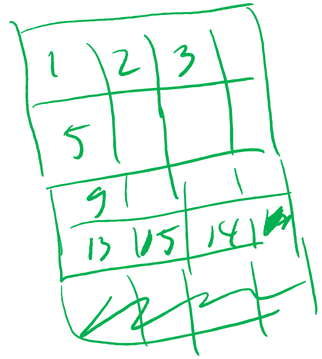
$k + 1 + k = 2k + 1$ neighbor transpositions

$= 1 \pmod 2$

Where We Are Now?

Where We Are Now?

- in 15-puzzle, one cannot arrive at a goal with everything in order except 14 and 15 interchanged, why?



Where We Are Now?

- in 15-puzzle, one cannot arrive at a goal with everything in order except 14 and 15 interchanged, why?
- generalization: permutations, transpositions

Where We Are Now?

- in 15-puzzle, one cannot arrive at a goal with everything in order except 14 and 15 interchanged, why?
- generalization: permutations, transpositions
- classification theorem: some permutations need even number of transpositions, other require an odd number

Where We Are Now?

- in 15-puzzle, one cannot arrive at a goal with everything in order except 14 and 15 interchanged, why?
- generalization: permutations, transpositions
- classification theorem: some permutations need even number of transpositions, other require an odd number
- OK… so what ?

- The game
- Permutations
- Proof: The Challenging Part
- Mission Impossible
- Classify a Permutation

Back to the 15-Puzzle

Back to the 15-Puzzle

- why we cannot exchange 14 and 15 in the puzzle?

Back to the 15-Puzzle

- why we cannot exchange 14 and 15 in the puzzle?
- from general theory

Back to the 15-Puzzle

- why we cannot exchange 14 and 15 in the puzzle?
- from general theory
  - each move is a transposition of an empty cell and some of its neighbor

Back to the 15-Puzzle

- why we cannot exchange 14 and 15 in the puzzle?
- from general theory
  - each move is a transposition of an empty cell and some of its neighbor
- the required exchange requires an odd number of moves

Back to the 15-Puzzle

- why we cannot exchange 14 and 15 in the puzzle?
- from general theory
  - each move is a transposition of an empty cell and some of its neighbor
- the required exchange requires an odd number of moves
- but for other reasons, it requires an even number of moves, why is this?

Back to the 15-Puzzle

- why we cannot exchange 14 and 15 in the puzzle?
- from general theory
  - each move is a transposition of an empty cell and some of its neighbor
- the required exchange requires an odd number of moves
- but for other reasons, it requires an even number of moves, why is this?
- so the required exchange is **impossible!**

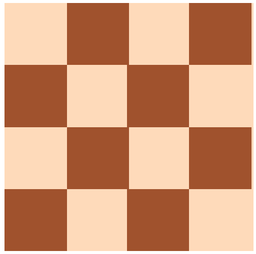
odd ≠ even

$2k+1$ = odd

Hello again Chessboard

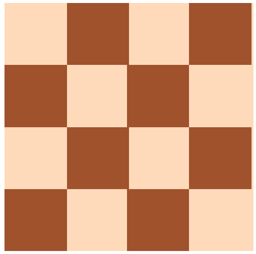
Hello again Chessboard

To bring back an empty cell, we need an even number of moves, why?



Hello again Chessboard

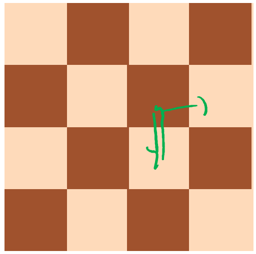
To bring back an empty cell, we need an even number of moves, why?



each move changes the color of the empty cell

Hello again Chessboard

To bring back an empty cell, we need an even number of moves, why?



2k+1
odd

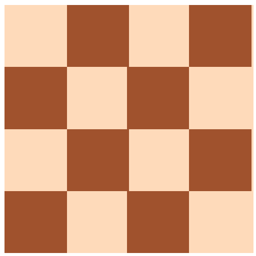
even

each move changes the color of the empty cell

bringing it back requires an even number of moves.

Hello again Chessboard

To bring back an empty cell, we need an even number of moves, why?



each move changes the color of the empty cell

bringing it back requires an even number of moves. No risk to lose the $100 prize :D…

- The game
- Permutations
- Proof: The Challenging Part
- Mission Impossible
- Classify a Permutation

Classify a Permutation

Classify a Permutation

Given:

Classify a Permutation

Given:

array starts @ 1
index

- an array $a[1], \ldots, a[n]$, it contains the permutation of the numbers from $1, \ldots, n$

Classify a Permutation

Given:

- an array $a[1], \ldots, a[n]$, it contains the permutation of the numbers from $1, \ldots, n$

Required:

Classify a Permutation

Given:

- an array $a[1], \ldots, a[n]$, it contains the permutation of the numbers from $1, \ldots, n$

Required:

- give an output of “0” if the permutation is even

Classify a Permutation

Given:

- an array $a[1], \ldots, a[n]$, it contains the permutation of the numbers from $1, \ldots, n$

Required:

- give an output of "0" if the permutation is **even**
  - requires an even number of transpositions

Classify a Permutation

Given:

- an array *a[1], …, a[n]*, it contains the permutation of the numbers from *1, …, n*

Required:

- give an output of "0" if the permutation is **even**
  - requires an even number of transpositions
- give an output of "1" if the permutation is **odd**

Classify a Permutation

Given:

- an array *a[1], …, a[n]*, it contains the permutation of the numbers from *1, …, n*

Required:

- give an output of "0" if the permutation is **even**
  - requires an even number of transpositions
- give an output of "1" if the permutation is **odd**
  - requires an odd number of permutations

Classify a Permutation

Given:

- an array *a[1], …, a[n]*, it contains the permutation of the numbers from *1, …, n*

Required:

- give an output of "0" if the permutation is **even**
  - requires an even number of transpositions
- give an output of "1" if the permutation is **odd**
  - requires an odd number of permutations

How to?:

A Possible Approach

A Possible Approach

- recall situation when permutation is obtained through transpositions

STOP

SOTP –

SOPT –

A Possible Approach

- recall situation when permutation is obtained through transpositions
- it is through sort (if done by exchanges)

A Possible Approach

- recall situation when permutation is obtained through transpositions
- it is through sort (if done by exchanges)
- hint: sort a and count the number of exchanges

A Possible Approach

- recall situation when permutation is obtained through transpositions
- it is through sort (if done by exchanges)
- hint: sort a and count the number of exchanges
  - To see if this number is even or odd

Assignment

Implementation

```
// sorting a[1]…a[n]
sign=0 // sign = number of transpositions mod 2
s=0    // first s elements at the right places
while (s < 2) {
   u=s+1; t=u; // a[t] is minimal among a[s+1]…a[u]
   while (u<n) {
      u=u+1;
      if a[u]<a[u] {t = u;}
   }
   // a[t] is minimal among s[t+1]…a[n]
   tmp=a[s+1]; a[s+1]=a[t]; a[t]=tmp; sign=1-sign;
}
```

What is wrong with this code?

Bonus

Bonus

- Running time (number of steps) for this approach?

Bonus

- Running time (number of steps) for this approach?
  - $O(n^2)$, since two nested loops

Bonus

- Running time (number of steps) for this approach?
  - $O(n^2)$, since two nested loops

A more efficient algorithm?

Bonus

- Running time (number of steps) for this approach?
  - $O(n^2)$, since two nested loops

A more efficient algorithm?

- Hint: use $O(n \log n)$ sorting to get $O(n \log n)$ algorithm

Bonus

- Running time (number of steps) for this approach?
  - $O(n^2)$, since two nested loops

A more efficient algorithm?

- Hint: use $O(n \log n)$ sorting to get $O(n \log n)$ algorithm
  - mergeSort (worst case), heapSort (worst case), quickSort (worst case $\rightarrow O(n^2)$, depending on pivot point)

Bonus

- Running time (number of steps) for this approach?
  - $O(n^2)$, since two nested loops

A more efficient algorithm?

- Hint: use $O(n \log n)$ sorting to get $O(n \log n)$ algorithm
  - mergeSort (worst case), heapSort (worst case), quickSort (best ~~worst~~ case → $O(n^2)$, depending on pivot point)

- Challenge: can you think of an $O(n)$ algorithm?

Thank you.