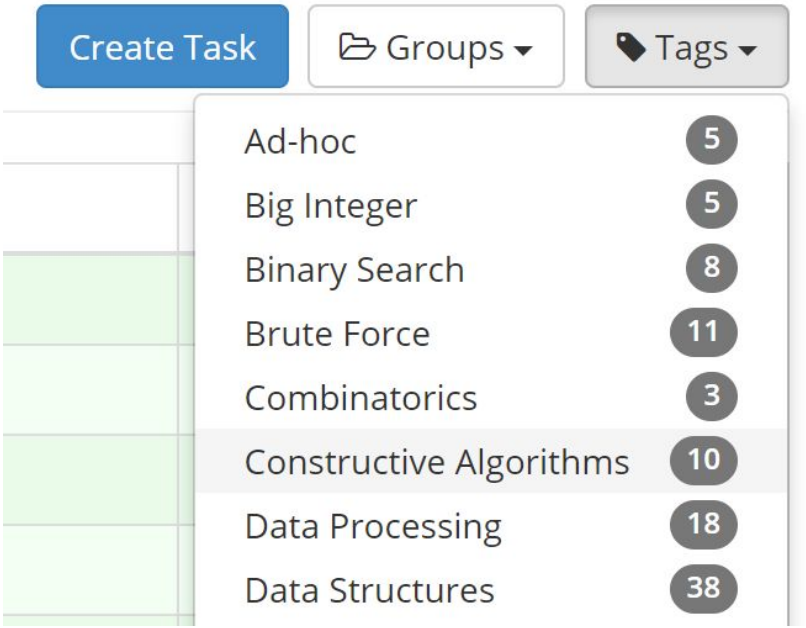
Constructive Algorithms

Percy Wong {percywtc}

What is it?



Constructive Algorithms

ID	Name
I0212	Utopia Divided
J144	Fair Santa Claus
J151	Inverse Problem
J161	Model Answer
J172	Card Game
J182	Rope
M1522	Gyeolhap
M1532	Inverse Problem 10
M1623	Bishop Puzzle
S132	Safe Storage
S134	Unfair Santa Claus
S163	Arithmetic Sequence

Constructive Problems?

Inverse Problem

J151 | Time Limit: 1.000 s | Memory Limit: 256 MB

If there are more than one valid sets, you can output any of them.

Model Answer

J161 | Time Limit: 1.000 s | Memory Limit: 256 MB

If there are multiple answers, you may output any one of them.

Card Game

J172 | Time Limit: 1.000 s | Memory Limit: 256 MB

If there are several solutions, output any.

Rope

J182 | Time Limit: 1.000 s | Memory Limit: 256 MB

Please help Alice to find any possible way to achieve so.

Bishop Puzzle

M1623 | Time Limit: 1.000 s | Memory Limit: 256 MB

If there are more than one solutions to the puzzle, output any.

Arithmetic Sequence

S163 | Time Limit: 1.000 s | Memory Limit: 256 MB

If there are more than one arrangement, output any one of them.

Constructive Problems...

- Usually give some requirements / constraints to be fulfilled
- You should construct any arrangement that satisfies the given rules
  - Permutations
  - Sequences
  - Matrices
  - Placements
  - ...
- Mostly interesting
- Often require thinking more than coding (standard algorithms)
- May have various correct solutions and “seemingly correct solutions”

How to approach them?

1. Solve some small cases manually / with the aid of programs
2. Observe patterns / relations between them
3. Making some “reasonable” guesses
4. Convince yourself that the guess is correct (or incorrect?)

Let's solve S163 together :)

The problem: given an integer **N**, output a permutation of 1..**N**, such that:

No any triples (**i**, **j**, **k**) where $1 \le i < j < k \le N$ such that $S_j - S_i = S_k - S_j$

Examples for **N = 6**:

Valid: **{3, 5, 4, 1, 2, 6}** and **{5, 1, 6, 3, 2, 4}**
Invalid: **{1, 4, 3, 2, 5, 6}** and **{6, 3, 4, 1, 5, 2}**

1. Solving small cases with programs

Write a simple program (not necessarily fast) to exhaust all possibilities

```cpp
#include<bits/stdc++.h>
using namespace std;

int N, S[200055];

bool valid() {
    for (int i = 0; i < N; i++)
        for (int j = i + 1; j < N; j++)
            for (int k = j + 1; k < N; k++)
                if (S[j] - S[i] == S[k] - S[j])
                    return false;
    return true;
}
```

```cpp
int main () {

    scanf("%d", &N);

    for (int i = 0; i < N; i++)
        S[i] = i + 1;

    do {
        if (valid()) {
            for (int i = 0; i < N; i++)
                printf("%d%c", S[i], (i == N - 1) ? '\n' : ' ');
        }
    } while (next_permutation(S, S + N));

    return 0;
}
```

1. Solving small cases with programs

Write a simple program (not necessarily fast) to exhaust all possibilities

```
4
1 3 2 4
1 3 4 2
2 1 4 3
2 4 1 3
2 4 3 1
3 1 2 4
3 1 4 2
3 4 1 2
4 2 1 3
4 2 3 1
```

```
5
1 5 3 2 4
1 5 3 4 2
2 1 4 5 3
2 4 1 5 3
2 4 3 1 5
2 4 3 5 1
2 4 5 1 3
3 1 2 5 4
3 1 5 2 4
3 1 5 4 2
3 5 1 2 4
3 5 1 4 2
3 5 4 1 2
```

```
6
1 5 3 2 6 4
1 5 3 4 2 6
1 5 3 4 6 2
1 5 3 6 2 4
1 5 6 3 2 4
2 1 6 4 5 3
2 6 1 4 5 3
2 6 4 1 5 3
2 6 4 3 1 5
2 6 4 3 5 1
2 6 4 5 1 3
3 1 2 5 6 4
3 1 5 2 6 4
3 1 5 4 2 6
3 1 5 4 6 2
3 1 5 6 2 4
3 5 1 2 6 4
3 5 1 4 2 6
3 5 1 4 6 2
3 5 1 6 2 4
3 5 4 1 2 6
```

```
8
1 5 3 2 7 6 4 8
1 5 3 2 7 6 8 4
1 5 3 7 2 6 4 8
1 5 3 7 2 6 8 4
1 5 3 7 4 2 8 6
1 5 3 7 4 2 8 6
1 5 3 7 4 8 2 6
1 5 3 7 4 8 6 2
1 5 3 7 6 2 4 8
1 5 3 7 6 2 8 4
1 5 3 7 6 8 2 4
1 5 3 7 8 4 2 6
1 5 3 7 8 4 6 2
1 5 7 3 2 6 4 8
1 5 7 3 2 6 8 4
1 5 7 3 4 2 8 6
1 5 7 3 4 8 2 6
1 5 7 3 4 8 6 2
1 5 7 3 6 2 4 8
1 5 7 3 6 2 8 4
1 5 7 3 6 8 2 4
1 5 7 3 8 4 2 6
```

```
9
1 9 5 3 2 7 6 4 8
1 9 5 3 2 7 6 8 4
1 9 5 3 7 2 6 4 8
1 9 5 3 7 2 6 8 4
1 9 5 3 7 4 2 8 6
1 9 5 3 7 4 8 2 6
1 9 5 3 7 4 8 6 2
1 9 5 3 7 6 2 4 8
1 9 5 3 7 6 2 8 4
1 9 5 3 7 6 8 2 4
1 9 5 3 7 8 4 2 6
1 9 5 3 7 8 4 6 2
1 9 5 7 3 2 6 4 8
1 9 5 7 3 2 6 8 4
1 9 5 7 3 4 2 8 6
1 9 5 7 3 4 8 2 6
1 9 5 7 3 4 8 6 2
1 9 5 7 3 6 2 4 8
1 9 5 7 3 6 2 8 4
1 9 5 7 3 6 8 2 4
1 9 5 7 3 8 4 2 6
1 9 5 7 3 8 4 6 2
1 9 5 7 8 3 4 2 6
1 9 5 7 8 3 4 6 2
```

2. Observe patterns / relations between them

Can you observe any patterns (or phenomena happened on most cases)?

```
4
1 3 2 4
1 3 4 2
2 1 4 3
2 4 1 3
2 4 3 1
3 1 2 4
3 1 4 2
3 4 1 2
4 2 1 3
4 2 3 1
```

```
5
1 5 3 2 4
1 5 3 4 2
2 1 4 5 3
2 4 1 5 3
2 4 3 1 5
2 4 3 5 1
2 4 5 1 3
3 1 2 5 4
3 1 5 2 4
3 1 5 4 2
3 5 1 2 4
3 5 1 4 2
3 5 4 1 2
```

```
6
1 5 3 2 6 4
1 5 3 4 2 6
1 5 3 4 6 2
1 5 3 6 2 4
1 5 6 3 2 4
2 1 6 4 5 3
2 6 1 4 5 3
2 6 4 1 5 3
2 6 4 3 1 5
2 6 4 3 5 1
2 6 4 5 1 3
3 1 2 5 6 4
3 1 5 2 6 4
3 1 5 4 2 6
3 1 5 4 6 2
3 1 5 6 2 4
3 5 1 2 6 4
3 5 1 4 2 6
3 5 1 4 6 2
3 5 1 6 2 4
3 5 4 1 2 6
```

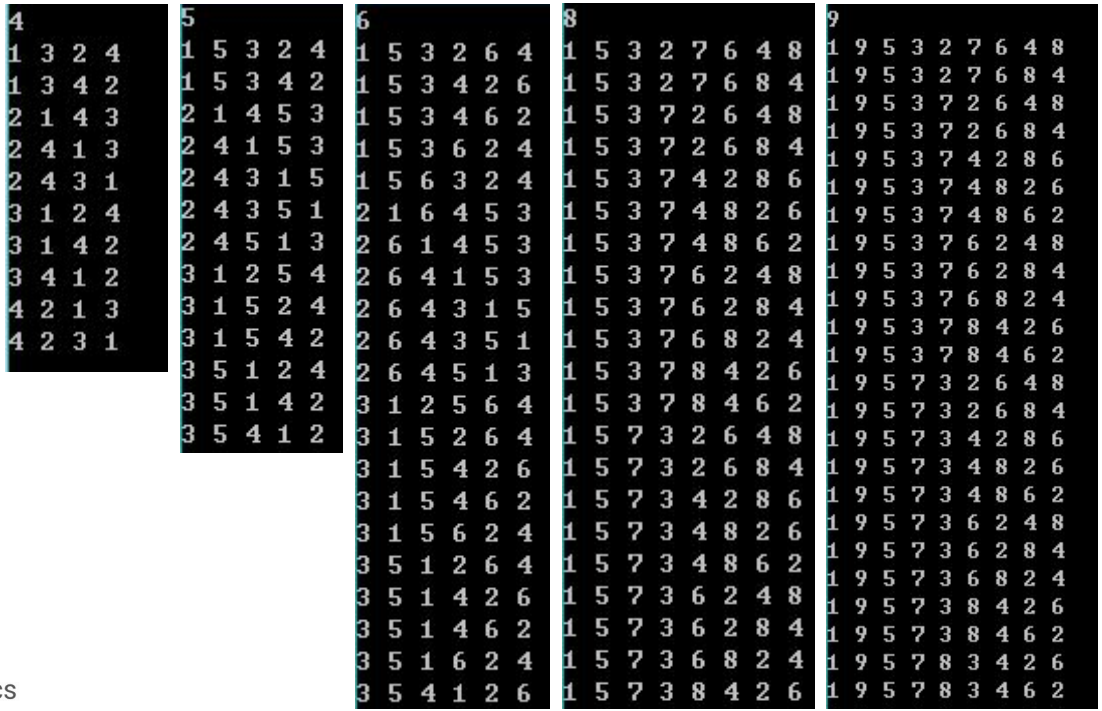
```
8
1 5 3 2 7 6 4 8
1 5 3 2 7 6 8 4
1 5 3 7 2 6 4 8
1 5 3 7 2 6 8 4
1 5 3 7 4 2 8 6
1 5 3 7 4 2 8 6
1 5 3 7 4 8 2 6
1 5 3 7 4 8 6 2
1 5 3 7 6 2 4 8
1 5 3 7 6 2 8 4
1 5 3 7 6 8 2 4
1 5 3 7 8 4 2 6
1 5 3 7 8 4 6 2
1 5 7 3 2 6 4 8
1 5 7 3 2 6 8 4
1 5 7 3 4 2 8 6
1 5 7 3 4 8 2 6
1 5 7 3 4 8 6 2
1 5 7 3 6 2 4 8
1 5 7 3 6 2 8 4
1 5 7 3 6 8 2 4
1 5 7 3 8 4 2 6
```

```
9
1 9 5 3 2 7 6 4 8
1 9 5 3 2 7 6 8 4
1 9 5 3 7 2 6 4 8
1 9 5 3 7 2 6 8 4
1 9 5 3 7 4 2 8 6
1 9 5 3 7 4 8 2 6
1 9 5 3 7 4 8 6 2
1 9 5 3 7 6 2 4 8
1 9 5 3 7 6 2 8 4
1 9 5 3 7 6 8 2 4
1 9 5 3 7 8 4 2 6
1 9 5 3 7 8 4 6 2
1 9 5 7 3 2 6 4 8
1 9 5 7 3 2 6 8 4
1 9 5 7 3 4 2 8 6
1 9 5 7 3 4 8 2 6
1 9 5 7 3 4 8 6 2
1 9 5 7 3 6 2 4 8
1 9 5 7 3 6 2 8 4
1 9 5 7 3 6 8 2 4
1 9 5 7 3 8 4 2 6
1 9 5 7 3 8 4 6 2
1 9 5 7 8 3 4 2 6
1 9 5 7 8 3 4 6 2
```

2. Observe patterns / relations between them

Can you observe any patterns (or phenomena happened on most cases)?

Numbers with the same parity are more likely to be staying together

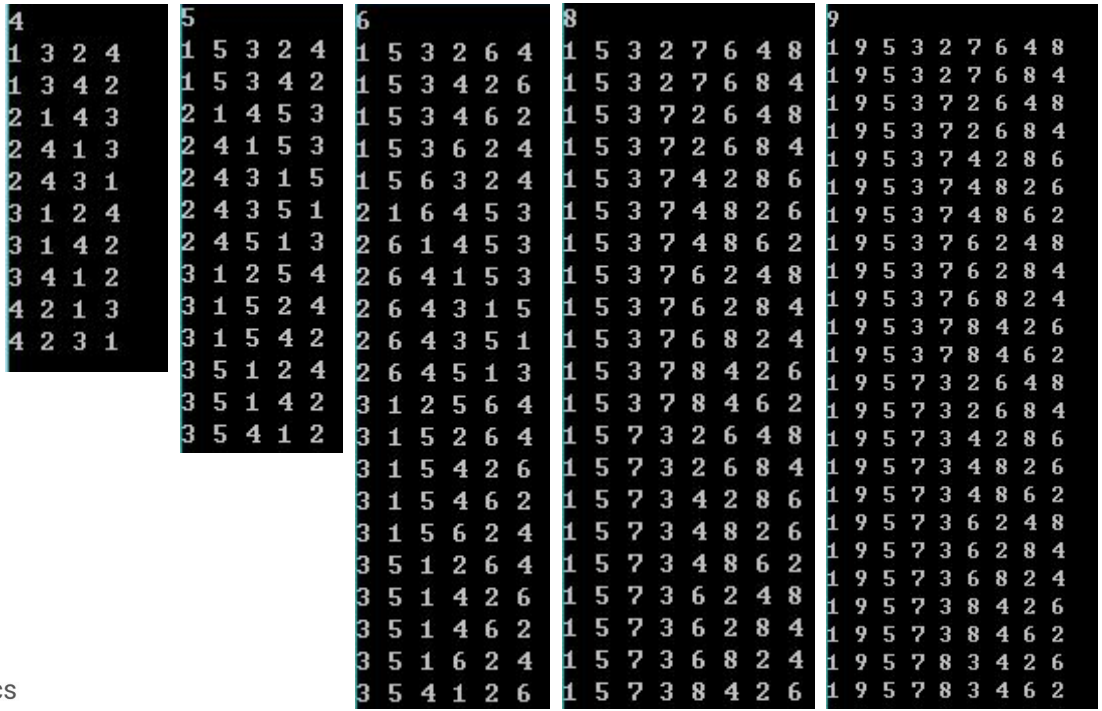


3. Making some “reasonable” guesses

Can you observe any patterns (or phenomena happened on most cases)?

Numbers with the same parity are more likely to be staying together

“Maybe” we should first divide the permutation into odds and evens?



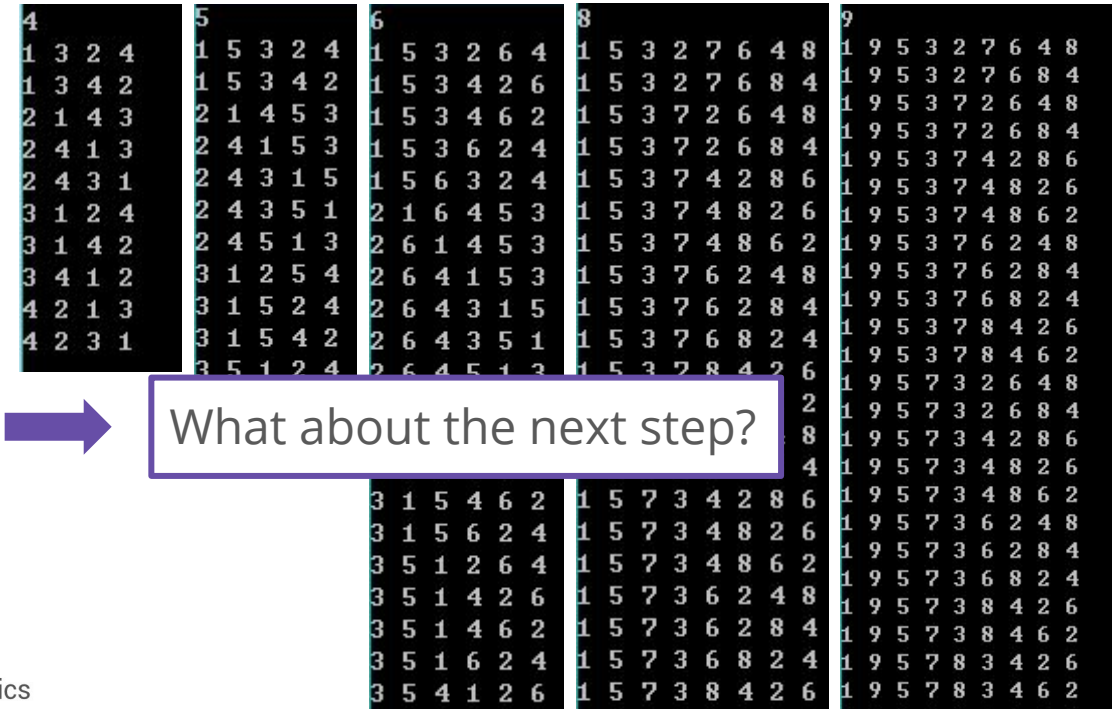
3. Making some "reasonable" guesses

Can you observe any patterns (or phenomena happened on most cases)?

Numbers with the same parity are more likely to be staying together

"Maybe" we should first divide the permutation into odds and evens?

What about the next step?

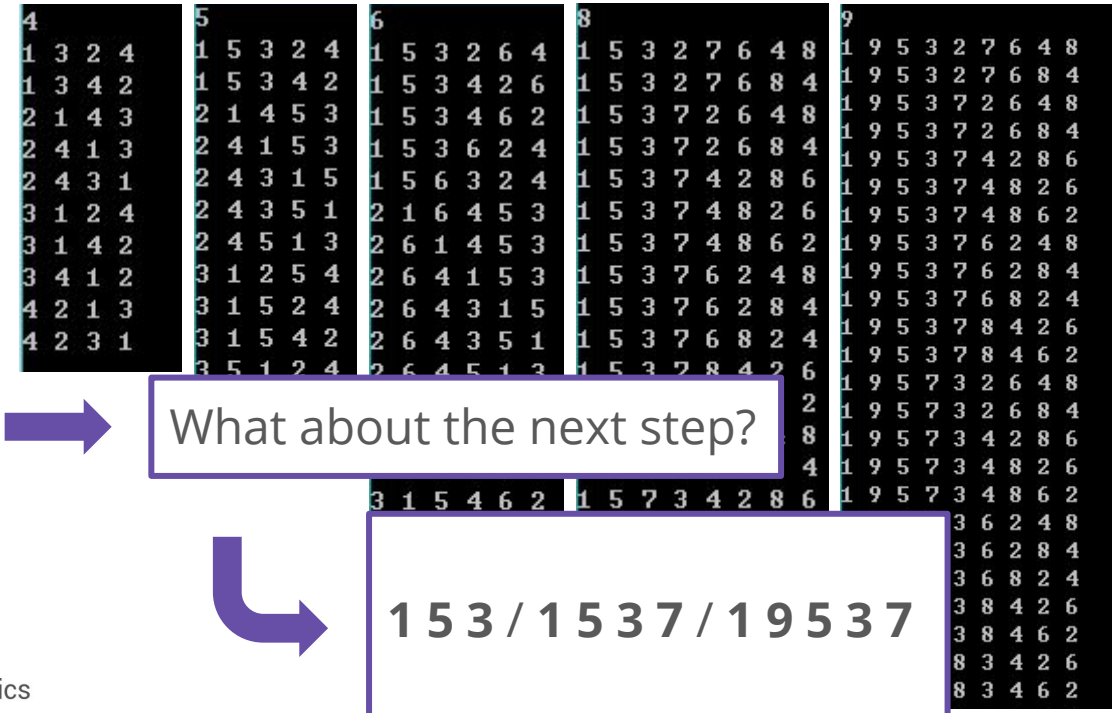


香港電腦奧林匹克競賽
Hong Kong Olympiad in Informatics

2. Observe patterns / relations between them

Can you observe any patterns (or phenomena happened on most cases)?

Numbers with the same parity are more likely to be staying together



"Maybe" we should first divide the permutation into odds and evens?

What about the next step?

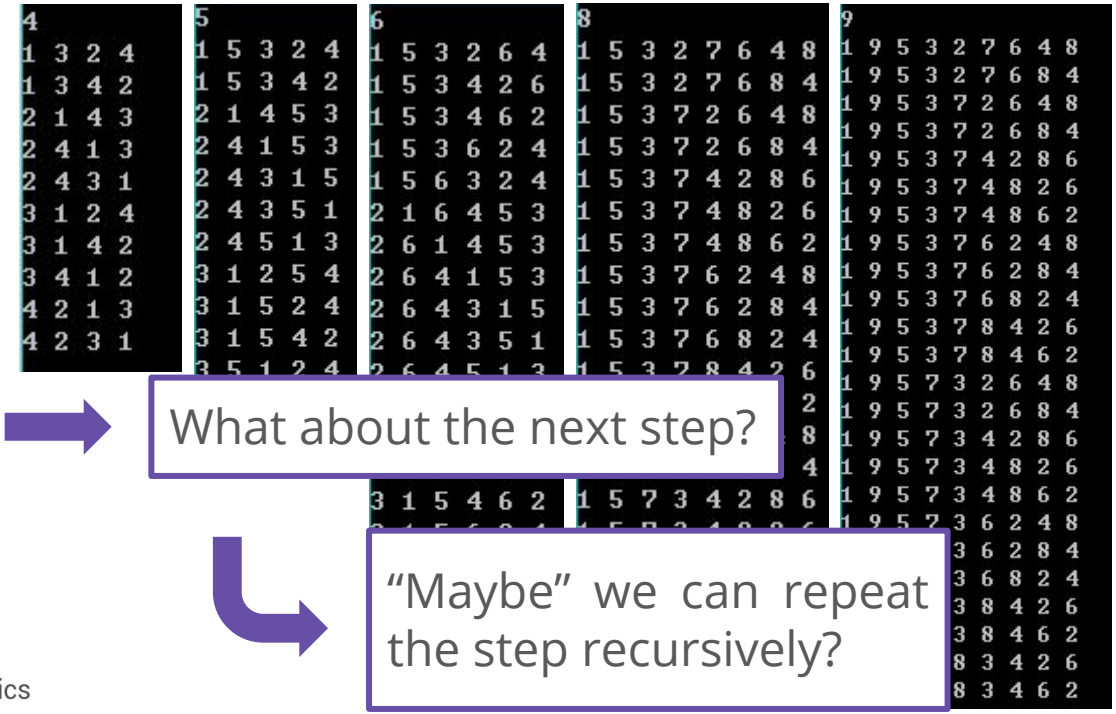
153 / 1537 / 19537

3. Making some "reasonable" guesses

Can you observe any patterns (or phenomena happened on most cases)?

Numbers with the same parity are more likely to be staying together

"Maybe" we should first divide the permutation into odds and evens?



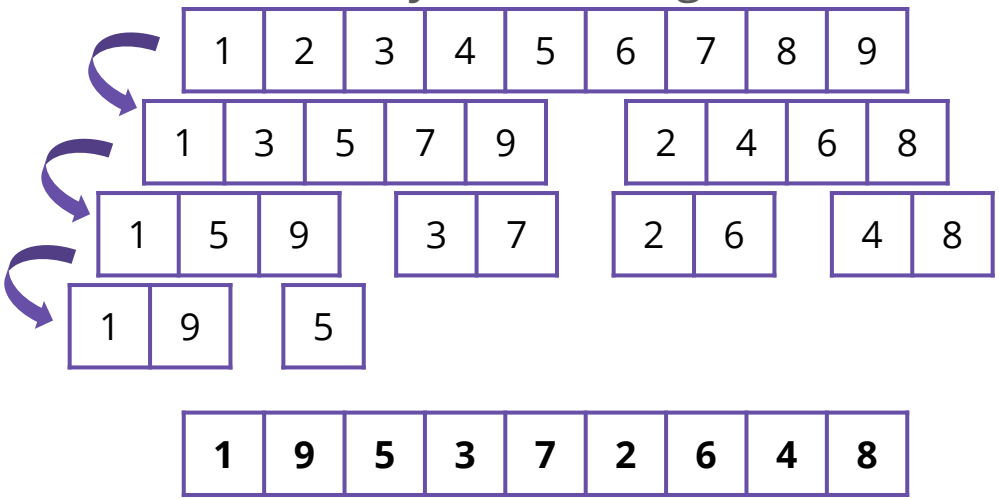
What about the next step?

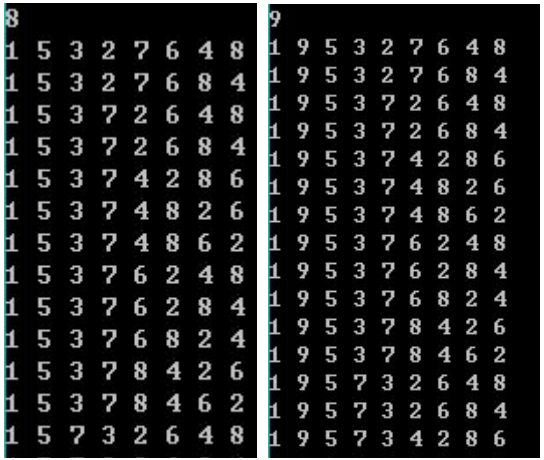
"Maybe" we can repeat the step recursively?

香港電腦奧林匹克競賽
Hong Kong Olympiad in Informatics

4. Convince yourself

So we are now trying to use the following solution:
Recursively divide the sequence into two groups by
alternatively distributing the numbers to the left and the right





During the break...

Two Problems... :)

- Given array **A** of **N** integers, find any subset that its sum is a multiple of **N**
  - **A = {3, 1, 4, 1, 5, 9}**, some solutions (indices in 1-based): **{1, 6}** or **{1, 3, 5}** or **{2, 3, 4}**

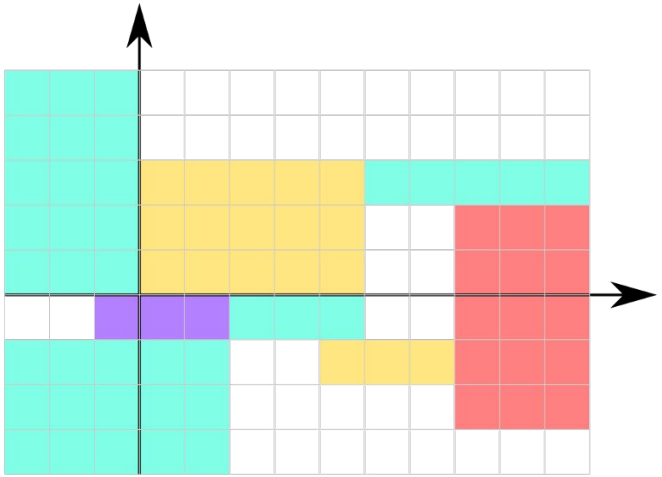
- Given **N**, find **N** consecutive integers that are all composite numbers
  - You may assume that the integers outputted can be infinitely large

Another Problem

[Codeforces 763B - Timofey and rectangles]

- Input
  - non-overlapping rectangles
  - integer coordinates
  - **odd lengths**

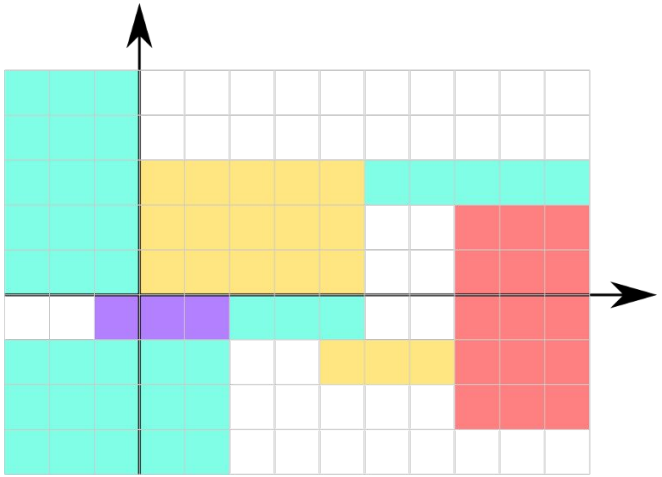
- Output
  - using 4 colors to color the rectangles
  - no touching rectangles share same color



Another Problem

[Codeforces 763B - Timofey and rectangles]

- Observations
  - Consider the bottom-left cell of rectangles
  - Only different parity in x-coordinate touch (Why?)
  - Same as the y-coordinate



- Idea
  - Coloring according to their parity
  - Combining x- / y-coordinate (How?)

Some more tips...

- Reducing the problem into smaller cases / lower dimensions
- Divide and Conquer
- Greedy Approach
- Binary notations

Some more tips...

- Use some random ideas and carefully analyze why are they incorrect
- Be careful on small / special cases
- Double-check the cases you solve manually / the exhaustion program
- Don’t think too much :)

Some more problems...

- Codeforces
  - Lesha and array splitting
  - From Y to Y
  - Dasha and Puzzle
  - Seating of Students
  - Divisibility
- AtCoder
  - Four Coloring
- CS Academy
  - Tennis Tournament
  - Manhattan Distances
  - BFS-DFS
  - Cyclic Shift (HARD)
- LS-PC Programming Challenge
  - Annoying Mathematics
  - Monorail
  - Gravitational Tetris