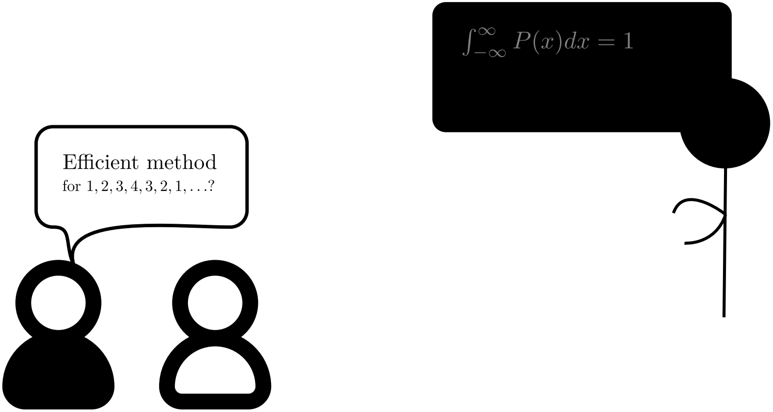
Logical treatment for the oscillatory sequence $1, 2, 3, 4, 3, 2, 1, 2, \ldots$ to find any term and a computer program to assist the operation

Damodar Rajbhandari, B.Sc.

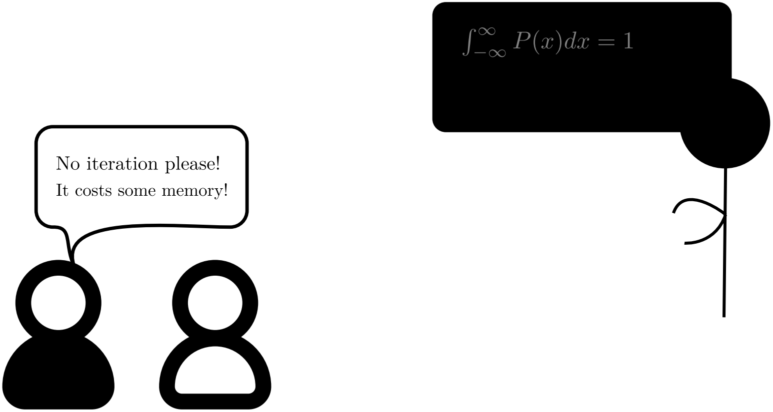
Theory of Complex Systems Division
Institute of Theoretical Physics
Jagiellonian University

Specialisation Seminar IV
20 May, 2021

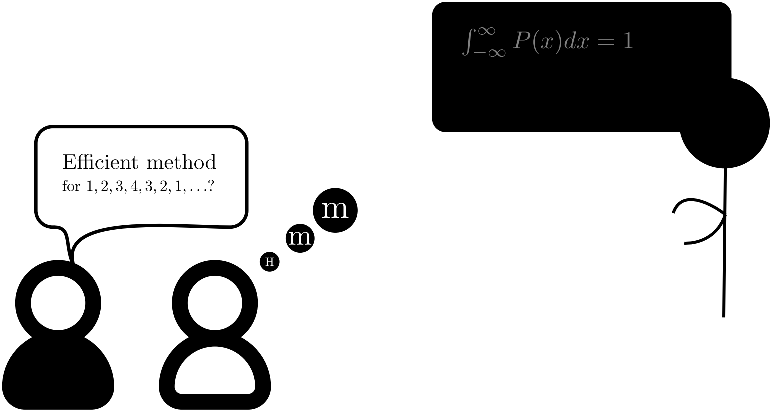
Let me take you back to my B.Sc. 2nd year



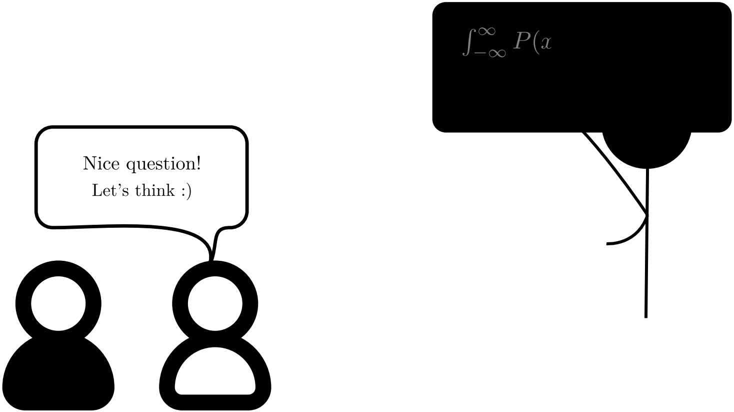
Let me take you back to my B.Sc. 2nd year



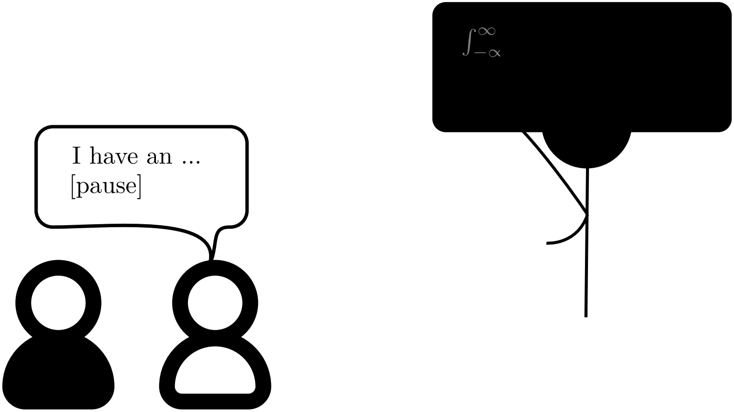
Let me take you back to my B.Sc. 2nd year



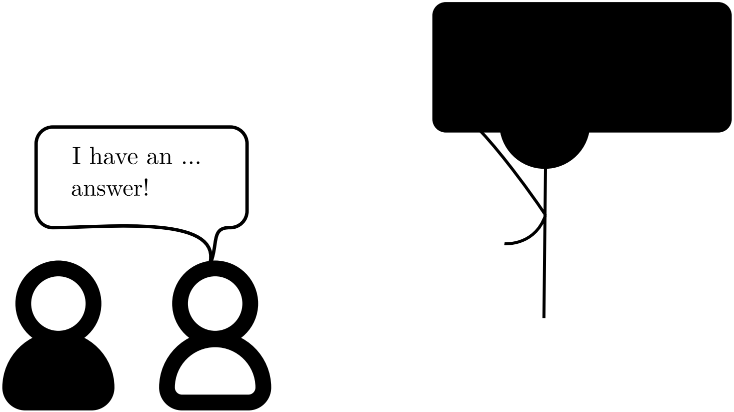
Let me take you back to my B.Sc. 2nd year



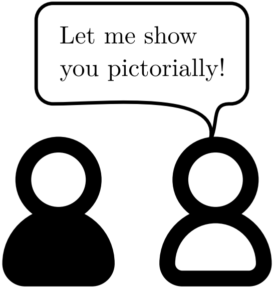
Let me take you back to my B.Sc. 2nd year

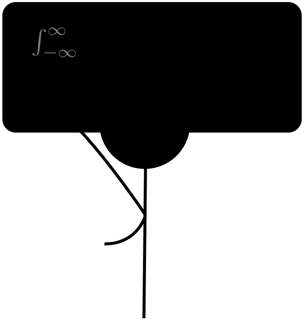


Let me take you back to my B.Sc. 2nd year

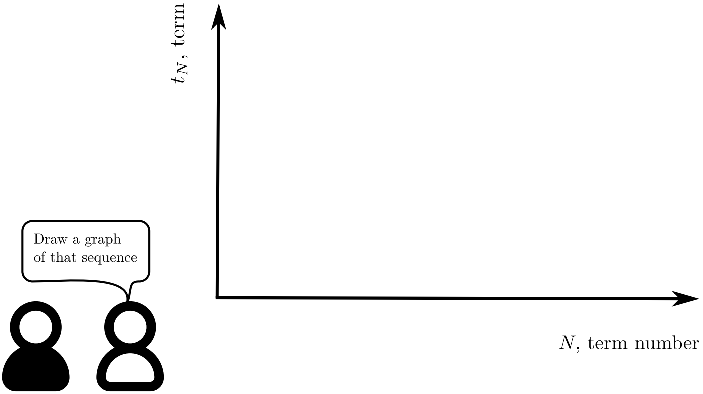


Let me take you back to my B.Sc. 2nd year

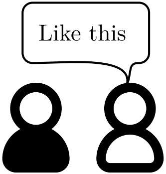


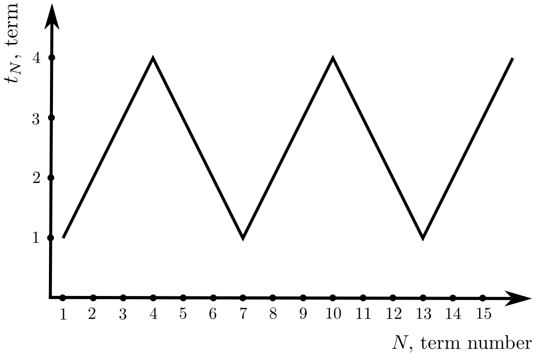


Let me take you back to my B.Sc. 2nd year

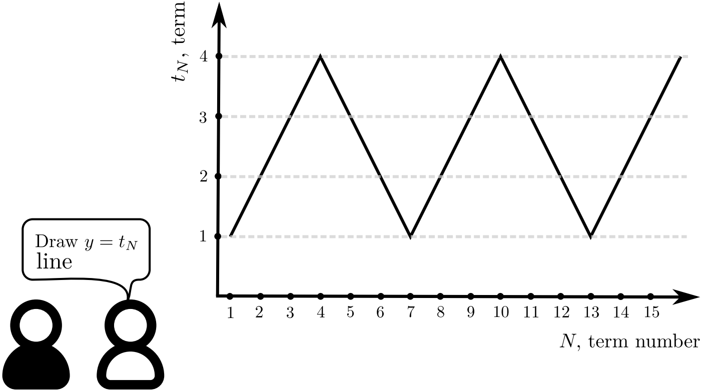


Let me take you back to my B.Sc. 2nd year

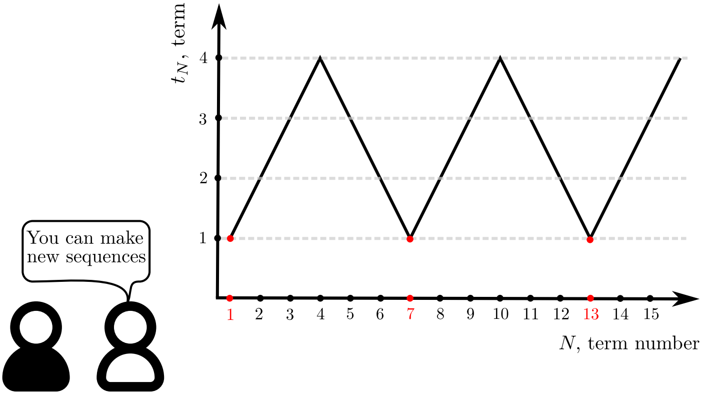




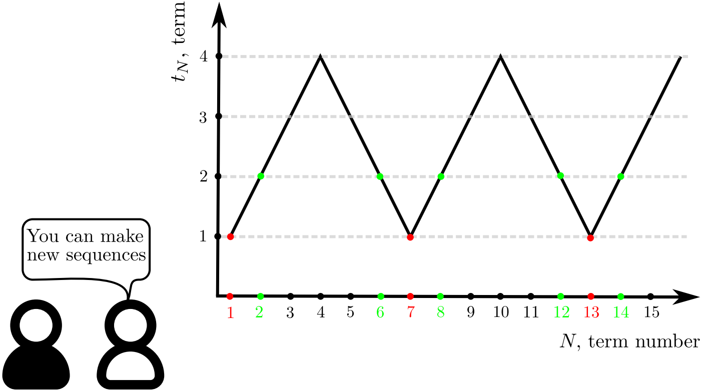
Let me take you back to my B.Sc. 2nd year



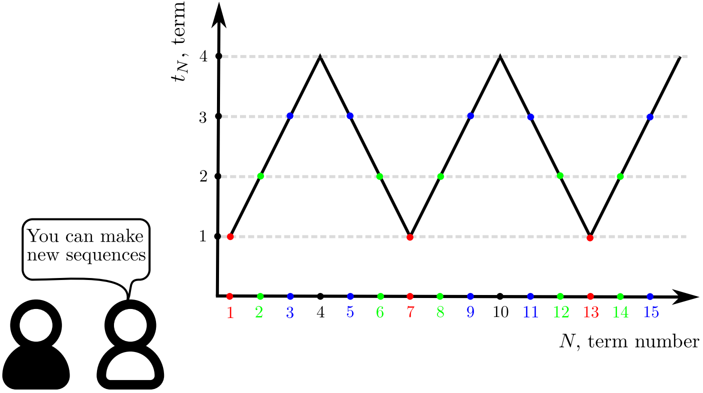
Let me take you back to my B.Sc. 2nd year



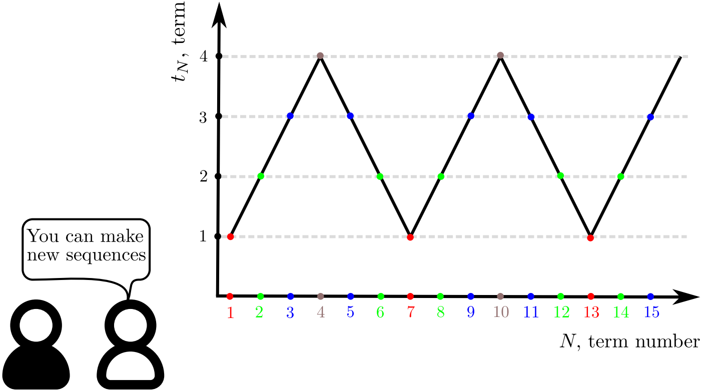
Let me take you back to my B.Sc. 2nd year



Let me take you back to my B.Sc. 2nd year



Let me take you back to my B.Sc. 2nd year



Now, we have four new arithmetic progressions

t_N	Lying on the $y = t_N$ line	General term
1	$1, 7, 13, 19, \ldots$ to i terms	t_i
2	$2, 6, 8, 12, 14, 18, \ldots$ to j terms	t_j
3	$3, 5, 9, 11, 15, 17, \ldots$ to k terms	t_k
4	$4, 10, 16, \ldots$ to l terms	t_l

Now, we have four new arithmetic progressions

t_N	Lying on the $y = t_N$ line	General term
1	$1, 7, 13, 19, \ldots$ to i terms	t_i
2	$2, 6, 8, 12, 14, 18, \ldots$ to j terms	t_j
3	$3, 5, 9, 11, 15, 17, \ldots$ to k terms	t_k
4	$4, 10, 16, \ldots$ to l terms	t_l

General terms formulae are

- $t_i = 6i - 5$ such that $i = 1, 2, 3, \ldots$
- $t_j = 3j$ when j is even
  $= 3j - 1$ when j is odd, such that $j = 1, 2, 3, \ldots$
- $t_k = 3k - 1$ when k is even
  $= 3k$ when k is odd, such that $k = 1, 2, 3, \ldots$
- $t_l = 6l - 2$ such that $l = 1, 2, 3, \ldots$

Logical part

- Notice t_i, t_j, t_k, and t_l are just no. of terms (N) in our original sequence. i.e. $t_x = N$ where $x \in i, j, k, l$

Logical part

- Notice $t_i, t_j, t_k,$ and t_l are just no. of terms (N) in our original sequence. i.e. $t_x = N$ where $x \in i, j, k, l$
- Revert the general terms

Logical part

- Notice t_i, t_j, t_k, and t_l are just no. of terms (N) in our original sequence. i.e. $t_x = N$ where $x \in i, j, k, l$
- Revert the general terms

For example,

$$t_i = 6i - 5 = N$$
$$\therefore i = \frac{N+5}{6}$$

Logical part

- Notice t_i, t_j, t_k, and t_l are just no. of terms (N) in our original sequence. i.e. $t_x = N$ where $x \in i, j, k, l$
- Revert the general terms

For example,

$$t_i = 6i - 5 = N$$
$$\therefore i = \frac{N+5}{6}$$

Condition: If $i \in \mathbb{N}$ then, the N^{th} term is 1.

Similarly,

$$t_j = 3j = N$$
$$\therefore j = \frac{N}{3}$$

Condition: If $j \in \mathbb{N}$ and j is even then, the N^{th} term is 2.

Similarly,

$$t_j = 3j = N$$
$$\therefore j = \frac{N}{3}$$

Condition: If $j \in \mathbb{N}$ and j is even then, the N^{th} term is 2.

$$t_j = 3j - 1 = N$$
$$\therefore j = \frac{N+1}{3}$$

Condition: If $j \in \mathbb{N}$ and j is old then, the N^{th} term is 2.

Similarly,

$$t_k = 3k - 1 = N$$
$$\therefore k = \frac{N+1}{3}$$

Condition: If $k \in \mathbb{N}$ and k is even then, the N^{th} term is 3.

$$t_k = 3k = N$$
$$\therefore k = \frac{N}{3}$$

Condition: If $k \in \mathbb{N}$ and k is old then, the N^{th} term is 3.

$$t_l = 6l - 2 = N$$
$$\therefore l = \frac{N+2}{6}$$

Condition: If $l \in \mathbb{N}$ then, the N^{th} term is 4.

Axioms (in-general)

- If all $(n - 1)$ tests fail then, the last test n must be true where, $n =$ total number of tests.
- If one test passed then, all other remaining tests must fail.

Axioms (in-general)

- If all $(n-1)$ tests fail then, the last test n must be true where, $n =$ total number of tests.
- If one test passed then, all other remaining tests must fail.

This is really helpful while implementing the logic we found earlier.

Computer Program & Closing Remarks

```python
#! /usr/bin/python3

N = int(input("Enter the value of N as you like: "))
if (N + 5) % 6 == 0:
  print("The {}th term is 1".format(N))
elif N % 3 == 0:
  if (N / 3) % 2 == 0:
    print("The {}th term is 2".format(N))
  else:
    print("The {}th term is 3".format(N))
elif (N + 1) % 3 == 0:
  if ((N + 1) / 3)% 2 == 0:
    print("The {}th term is 3".format(N))
  else:
    print("The {}th term is 2".format(N))
else:
  print("The {}th term is 4".format(N))
```