

# HoP101: Session 2

## Follow-up and Ideas on Project

Mrigank Pawagi

Indian Institute of Science

ACM-W Student Chapter, Summer 2023



Would anybody like to take notes for this session?



# Merge Sort

This is based on our **merging** algorithm from the last session.  
Suppose we have to sort a list of  $n$  numbers.

- How do we sort a list of  $\lfloor n/2 \rfloor$  numbers?  
If we could do that, we can simply merge the two sorted lists!
- $n$  is an abstract quantity! We could apply the same process to any  $n$ , including  $\lfloor n/2 \rfloor$ .
- Do we need to worry about termination?



Any ideas?



# Tower of Brahma

How do we move  $n$  disks from pole  $A$  to pole  $C$ ?

- How do we move  $n - 1$  disks from pole  $A$  to pole  $C$ ?

Suppose we could do that. How does it help?



If we could do that, we could

- Move  $n - 1$  disks from pole  $A$  to pole  $B$ . We know how to do this, since there is conceptual symmetry between the poles.
- Move the  $n$ th disk from pole  $A$  to pole  $C$ .
- Move  $n - 1$  disks from pole  $B$  to pole  $C$ . Again, we know how to do this.



# Tower of Brahma

Again,  $n$  is an **abstract** quantity!

We could apply the same process to any  $n$ , including  $n - 1$ , as long as we know when to stop.



# More on Loops

All loop constructs are equivalent because they can be **converted** to each other. At the lower levels of programming, there is only one way to loop: **jumps** and **conditions**.

- while (we saw this last time)
- for
- for-each (not in Python)
- do-while (not in Python)
- do-until (not in Python)

We will see how all of these can be converted to while loops.





## Discussion on the model

- Parameters
- Constraints
- Assumptions



# Ideas on Project

Ideas on implementation

Let us start by **decomposition**.



- Please make a folder and keep all your files in it.
- Although IDLE works fine, I would recommend using VSCode.
- We will use git only towards the end so don't worry about it for now.

