Introduction to Data Structures

By

Suvam Basak

Email: suvambasak22@iitk.ac.in

Web: suvambasak.github.io

What is Data Structures?

$$-4x^0 + 7x^1 + 6x^2$$

	0	1	2
Poly	-4	7	6

int poly [3];

- Data Structures (DS) is logical and mathematical representation of data memory
  - Logical: Student info, Employee info
  - Mathematical: Polynomials

```python
from dataclasses import dataclass

@dataclass
class Car:
    color: str
    manufacturer: str
    top_speed_km: int
```

What is Data Structures?

- Structured representation of data elements based on their logical relationships

- Data Structures mainly deals with:
  - How efficiently data could be stored and organized
  - How efficiently data could be retrieved and manipulated

What is Data Structures?

- Structured representation of data elements based on their logical relationships

- Data Structures mainly deals with:
  - How efficiently data could be stored and organized
  - How efficiently data could be retrieved and manipulated

Program = Data Structure + Algorithm

<mark>Algorithm</mark>: GCD(m, n)?

- Largest K that divides both m and n
- GCD(8, 12) = 4
- GCD(18, 25) = 1
- GCD(14, 63) = ?
- How to compute GCD with python?

GCD(14, 63) = ?

Factors of 14

- 1
- 2
- 7
- 14

Factors of 63

- 1
- 3
- 7
- 9
- 21
- 63

GCD(14, 63) = ?

Factors of 14

- 1
- ~~2~~
- 7
- ~~14~~

Factors of 63

- 1
- ~~3~~
- 7
- ~~9~~
- ~~21~~
- ~~63~~

Euclid's algorithm

- If d divides both m, n and m > n
- Then m = ad, n = bd
- So (m-n) = (ad − bd)
  - (m-n) = (a − b) x d
- Therefore, d divides both (m-n) and (a-b)
- So, GCD(m,n) = GCD(n, m-n)

GCD(63, 14) = ?

- GCD(14, 63-14) => GCD(14, 49)
- GCD(14, 49-14) => GCD(14, 35)
- GCD(14, 35-14) => GCD(14, 21)
- GCD(14, 21-14) => GCD(14, 7)
- GCD(7, 14-7) => GCD(7, 7) => **7**

Data Structure = Organized data + Operations

- How would you store the DOB of Aadhaar database?

DOB of Aadhaar database

```
int dob[3];                      int16_t dob;
```

DOB of Aadhaar database

`int dob[3];`

- Size of int: 4 Bytes

- Size of dob[3]: (4x3) = 12 Bytes

- Size of 1.4 Billion dob[3]: 15.7 GB

`int16_t dob;`

- Size of int: 2 Bytes

- Size of 1.4 Billion dob: 2.7 GB

Data Structure = Organized data + Operations

Concatenation of bits

- 10/07/23
- 1010/111/10111

01010 0111 0010111

Left/Right bit shift

- Empty: 000000000000000
- Add date: 0000000000001010
- Add month: 0000000010100111
- Add year - 0101001110010111