

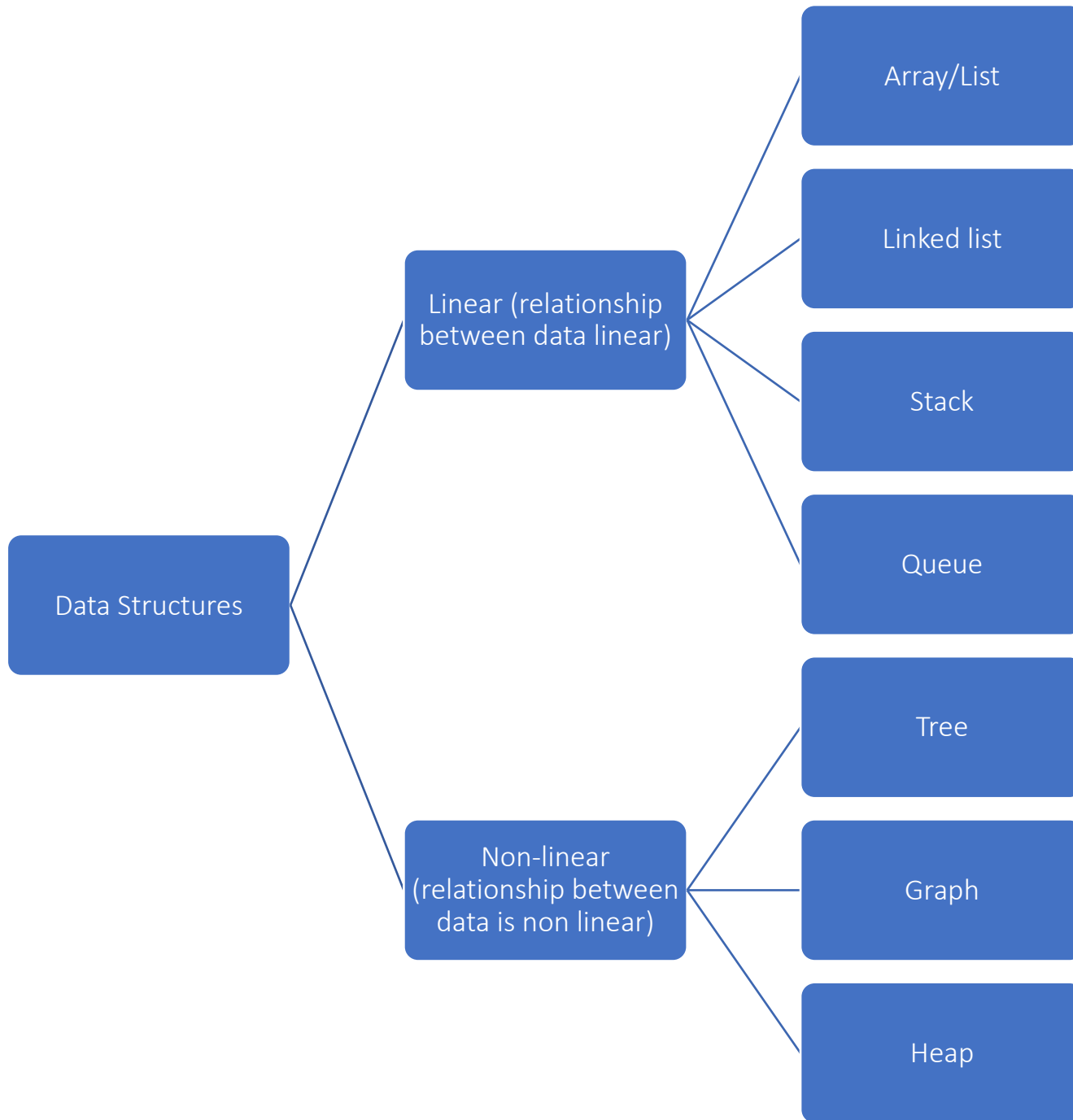
Data Structures

By

Suvam Basak

Email: suvambasak22@iitk.ac.in

Web: suvambasak.github.io



What is the difference

Array

1. It can only consist of value of same type
2. It can directly handle arithmetic operations
3. We need to import the array before work with the array
4. It is a more compact in memory size comparatively list
5. It is suitable for storing shorter sequence of data items
6. We can print the entire list without using explicit looping
7. It must contain either all nested elements of same size

List

1. The list can store the value of different types
2. The list cannot handle the direct arithmetic operations
3. The lists are the build-in data structure, so we don't need to import it
4. It consumes a large memory
5. It is suitable for storing the longer sequence of the data item
6. We can print the entire list using explicit looping
7. It can be nested to contain different types of elements

Array in Python

```
suvam@KD-304G: ~  
File Edit View Search Terminal Help  
Python 3.11.5 (main, Sep 11 2023, 13:54:46) [GCC 11.2.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>>  
>>> from array import *  
>>> arr = array('l', [1, 2, 3, 4, 5])  
>>> arr.buffer_info()  
(140696061016080, 5)  
>>>
```

Type code	C Type	Python Type	Minimum size in bytes	Notes
'b'	signed char	int	1	
'B'	unsigned char	int	1	
'u'	wchar_t	Unicode character	2	(1)
'h'	signed short	int	2	
'H'	unsigned short	int	2	
'i'	signed int	int	2	
'I'	unsigned int	int	2	
'l'	signed long	int	4	
'L'	unsigned long	int	4	
'q'	signed long long	int	8	
'Q'	unsigned long long	int	8	
'f'	float	float	4	
'd'	double	float	8	

Operation complexity

	Sorted Array (Ase)	Sorted Array (Des)	Sorted Array (Ase)	Unsorted Array	Unsorted Linked List	Sorted Double Linked List	Min Heap
Delete min	$O(n)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(1)$	$O(\log n)$
Delete n-th min	$O(n)$	$O(n)$	$O(n)$	$O(n^2)$	$O(n^2)$	$O(n)$	$O(n \log n)$
Find element (search)	$O(\log n)$	$O(\log n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
Key Decrement (ptr given)	$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$	$O(\log n)$
Delete element (ptr given)	$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(n)$	$O(1)$	$O(\log n)$
Insert element	$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$	$O(\log n)$