



ХИМИКОТЕХНОЛОГИЧЕН И МЕТАЛУРГИЧЕН УНИВЕРСИТЕТ - СОФИЯ

Алгоритми и структури данни

**лектор: доц. д-р инж. Атанас Атанасов
Катедра “Информатика”**

Лекция 8

СТУКТУРИ ОТ ДАННИ

СТРУКТУРИ ОТ ДАННИ

□ Въведение

□ Видове структури от данни

- Примитивни типове данни
- Непримитивни типове данни
- Линейни и нелинейни типове данни
- Хомогенни и нехомогенни типове данни
- Статични и динамични типове данни

Структури от данни - Въведение

- Структурите от данни са един от най-фундаменталните предмети в областта на компютърните науки.
- Всеки програмист трябва да има задълбочени познания в областта на на структурите от данни, за да може да създава ефективен софтуер и приложения.
- **Какво представлява структурата от данни и защо структурите от данни са толкова важни?**
 - Структурите от данни са множество от данни, които са организирани на основата на логически и математически закони (Wiki) и функции за обработка на тези данни.
 - Изборът на правилните структури от данни правят дадена програма по-ефективна, тъй като се спестява памет и време за изпълнение.
 - Данните биват групирани по определен начин, за да се улесни достъпът до тях и управлението им. За различни задачи са подходящи различни структури.

Структури от данни - Въведение

Структурите от данни могат да бъдат дефинирани като “начинът, по който данните се организират и съхраняват в паметта на компютъра”

Структурите от данни са набор от правила, които ни казват как да съхраняваме данни. Те правят достъпа до данни по-ефективен, и намаляват сложността при намирането и модифицирането на данни.

Структурите от данни са абстракция на и опростяване на реалността при решаване на конкретен проблем (задача). При тази абстракция се подбират данни, които са съществени за решаване на проблема и се игнорират други, които са несъществени.

Например при обработка на списъци на служители във фирма са важни имената, годините, стажа и заплатите, а не цвят на очите и косата, ръст, тегло и др.

Структури от данни - Въведение

- Пример с използване на речник. Добрият речник обикновено съдържа десетки хиляди думи с тяхното значение и големината му понякога е над хиляда страници. Нека предположим, че трябва да намерим определението на думата „информатика“. Как ще разберем значението ѝ?
- Първо, ще отворим страницата, в която има всички думи, започващи с „ин“ и след това ще потърсим нашата дума „информатика“, която е зададена по азбучен ред. Това прави търсенето на желаната дума много по-лесно благодарение на азбучния ред на думите, зададен от редакторите и издателите на речника.
- Издателите или редакторите са структурирали или организирали думите (или данните) в определен ред. Ако това не съществуваше и думите не са групирани по азбучен или друг ред, то търсенето на значението на дадена дума би ни отнело часове, може и да е безрезултатно.

Видове структури от данни

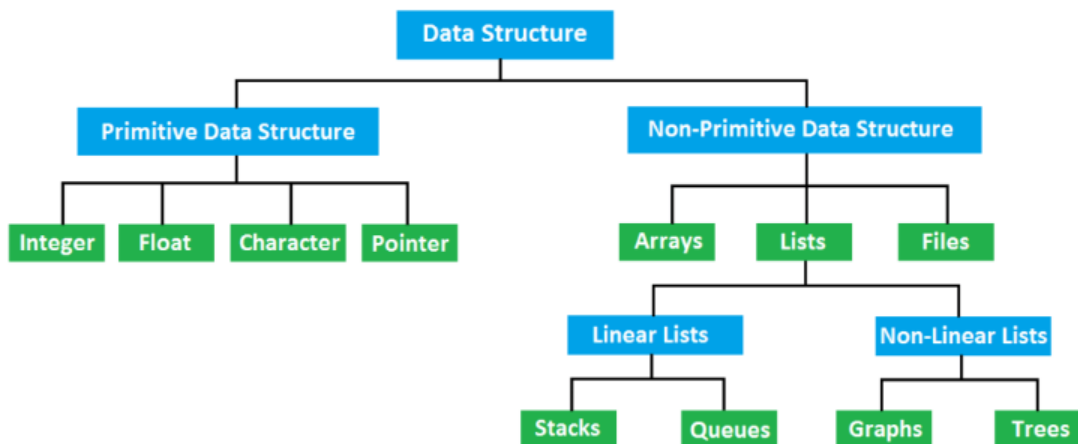
Типове данни

Типът на данните е атрибут, който указва на компилатора или интерпретатора каква памет да заделни за дадена променлива или константа.

Типовете данни са:

Примитивни или първични и включват целочислен тип (int), реален тип (float, double), символен (char), булев (bool) и типът указател (pointer)

Непримитивните или композитните типове включват записи (record, struct), масиви (array), файлове (file), линейни списъци (lists, stacks, queues), нелинейни списъци (дървета (trees), графи (graphs)) и др.



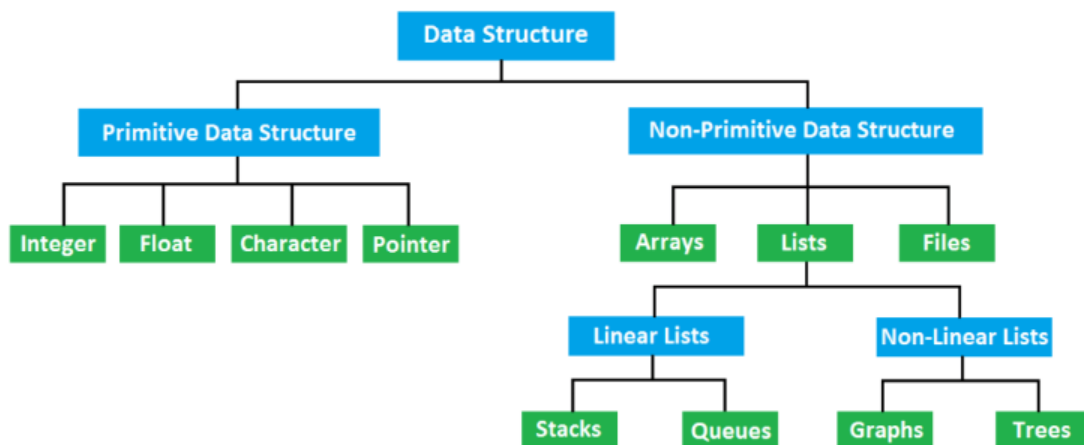
Видове структури от данни

Типове данни

Линейни или нелинейни, описващи дали елементите от данни са подредени в подредена последователност, например с масив, или в неподредена последователност, например граф.

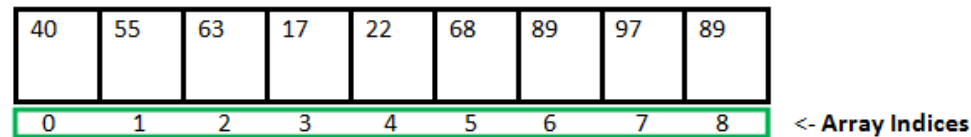
Хомогенни или нехомогенни, описващи дали всички данни са от един и същи тип или не. Например - при масивът всички елементи са от един тип, но записът е колекция от различни типове данни.

Статични или динамични, описващи дали данните имат фиксирани размери и място в паметта (зададено при компилирането) или данните се генерират динамично по време на изпълнение на програмата. Такива данни имат променлива структура и място в паметта на компютъра.



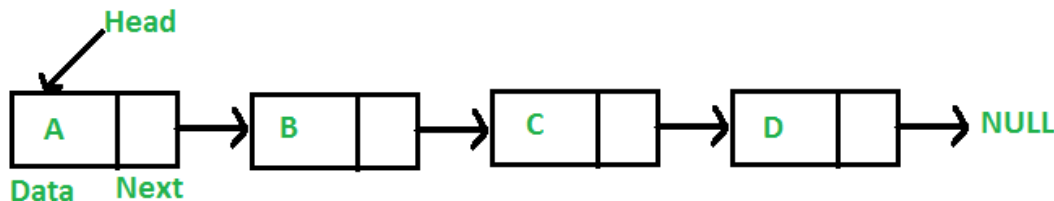
Видове структури от данни

Масивът съхранява колекция от елементи в съседни места на паметта. Елементите му са от същия тип данни, така че позицията на всеки елемент може лесно да бъде изчислена или извлечена. Масивите могат да бъдат с фиксирана или с променлива дължина.



Array Length = 9
First Index = 0
Last Index = 8

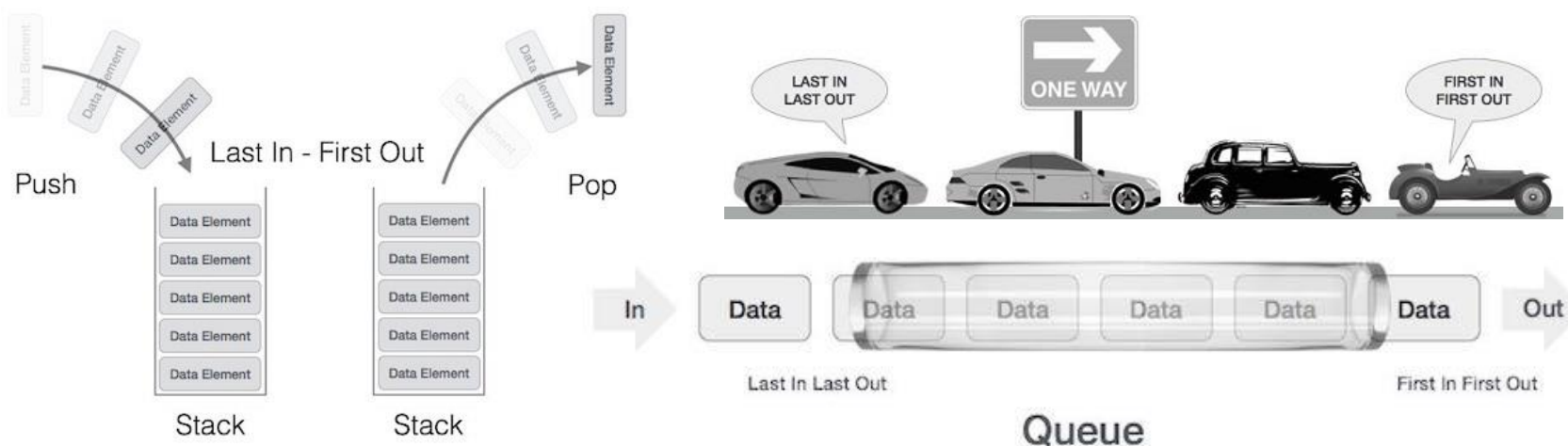
Свързаният списък съхранява колекция от елементи в линеен ред. Всеки елемент или възел в свързан списък съдържа данни, както и указател или връзка към следващия елемент в списъка. Свързаният списък е линейна структура от данни, където всеки възел е отделен обект. И за разлика от масива е гъвкав (динамичен) по своята същност.



Видове структури от данни

Стекът е линейна структура от данни, която следва определен ред, в който се извършват операциите над данните му. Данните се поставят и извличат от стека на основата на подредба от типа LIFO (Last In First Out) или FILO (First In Last Out). Така че достъпът до данните е свързан с тяхната подредба.

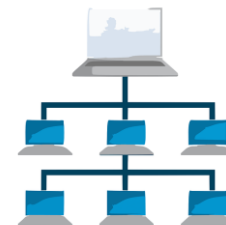
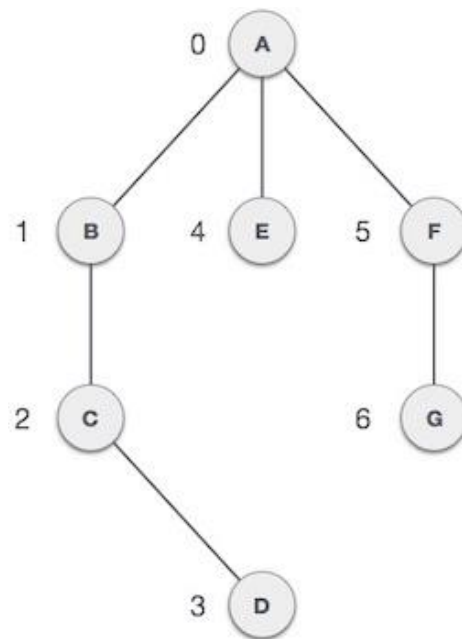
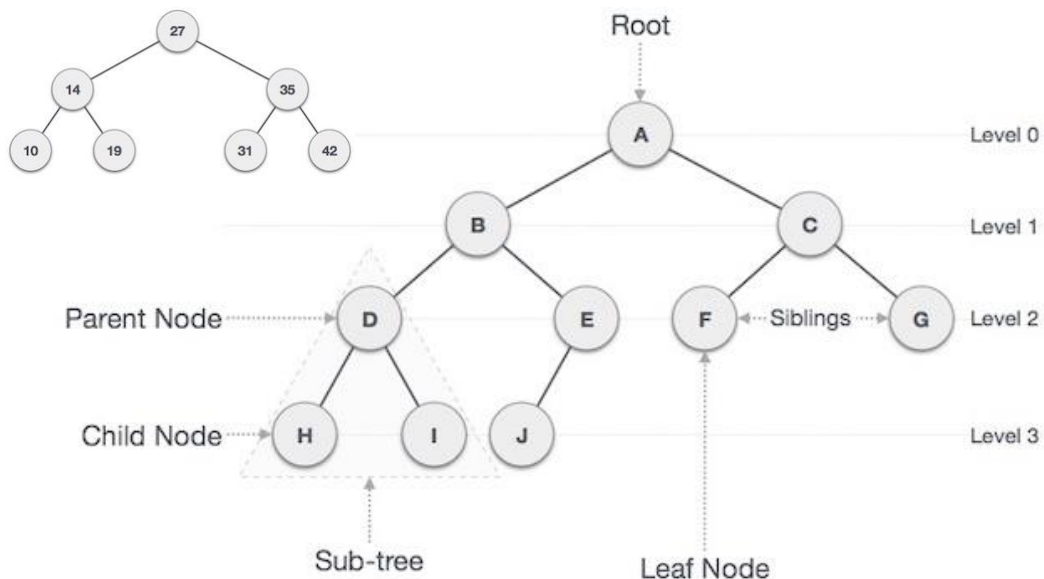
Опашката е линейна структура, която следва определен ред, в който се извършват операциите. Подредбата им е (First In First Out). Добър пример за опашка е всяка опашка от потребители за някакъв ресурс, при който потребителят, който е дошъл първи, е обслужен първи.



Видове структури от данни

Дървото съхранява колекция от елементи по абстрактен, йерархичен начин. Всеки възел е свързан с други възли и може да има множество под-стойности, известни също като деца.

Графът съхранява колекция от елементи по нелинеен начин. Графите са съставени от краен набор от възли, известни също като върхове *(vertex/vertices), и линии, които ги свързват, известни също като ръбове (edges). Те са полезни за представяне на реални системи като компютърни мрежи и др.



Видове структури от данни

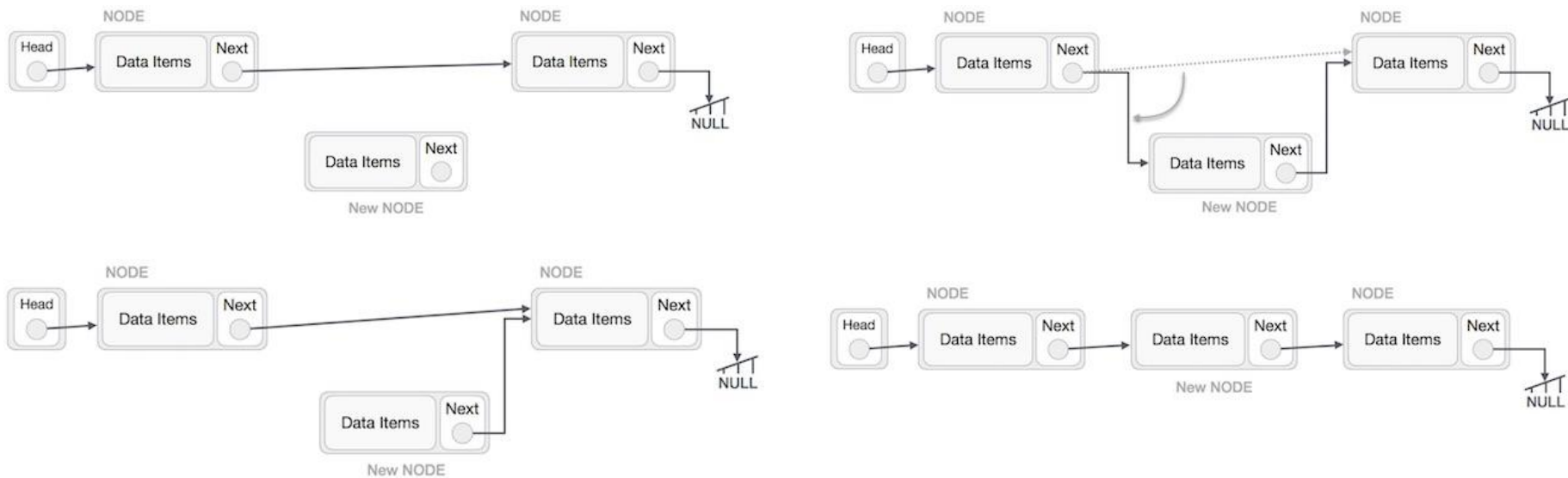
Структурата или записът е дефиниран от потребителя тип данни, който може да се използва за групиране на елементи от евентуално различни типове в един тип, вкл. и други структури/записи.

```
struct Person
{
    char name[50];
    int age;
    float salary;
};
```

Файлът е колекция от различни записи (или данни). Този тип структура от данни обикновено се използва за поддържане или управление на огромно количество данни, които не се намират в оперативната памет на компютъра, а са на харддиска или друг носител. Файловете ни позволяват да работим, обработваме, достъпваме, извличаме и дори модифицираме данните ефективно и бързо.

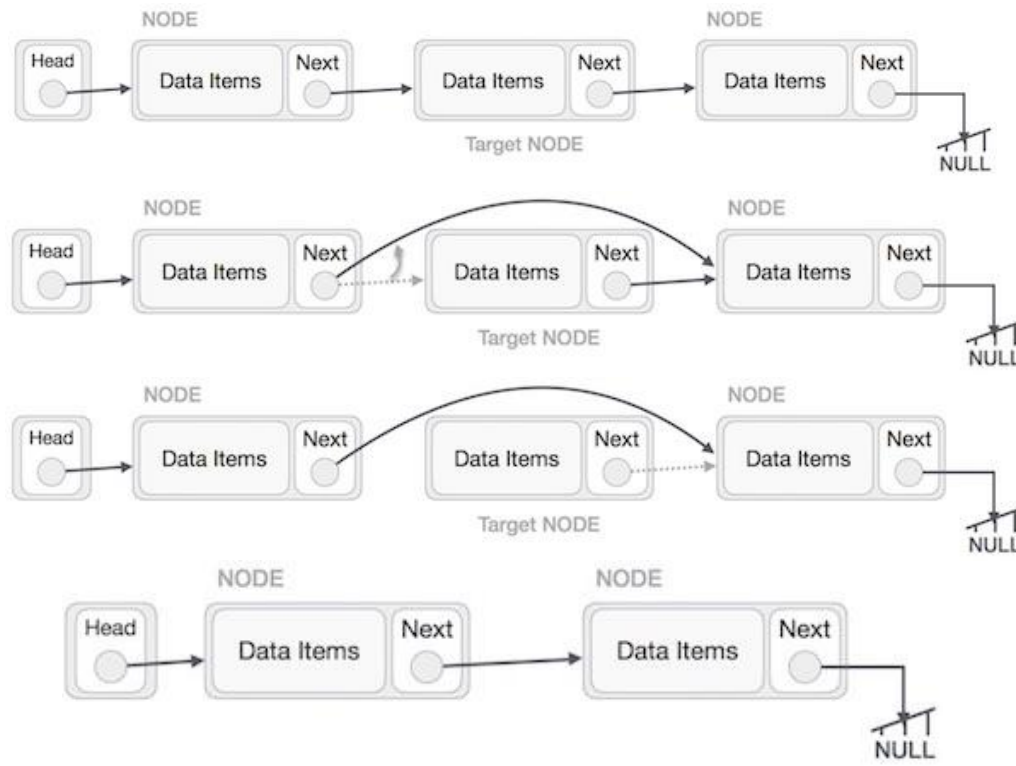
Операции със списъци - добавяне

- **Insertion** – добавя елемент в началото на списъка.
- **Deletion** – Изтрива елемент в началото на списъка.
- **Display** – Показва пълния списък.
- **Search** – Търси елемент с помощта на дадения ключ..
- **Delete** – Изтрива елемент с помощта на дадения ключ.



Операции със списъци - изтриване

- **Insertion** – добавя елемент в началото на списъка.
- **Deletion** – Изтрива елемент в началото на списъка.
- **Display** – Показва пълния списък.
- **Search** – Търси елемент с помощта на дадения ключ..
- **Delete** – Изтрива елемент с помощта на дадения ключ.



Операции с графи

- **Add vertex** – добавя възел в графа.
- **Add edge** – добавя ръб (връзка) между два възела .
- **Display vertex** – Показва възлите на графа.
- **Traversal** – обхождане на граф.

Правило 1 - Посетете съседния непосетен връх. Маркирайте го като посетен. Покажете го. Поставете го в стека.

Правило 2 - Ако не бъде намерен съседен връх, извадете го от стека. (Ще се получат всички върхове от стека, които нямат съседни върхове.)

Правило 3 - Повторете правило 1 и правило 2, докато стекът се изпразни.

