Двоични дървета

доц. д-р Нора Ангелова

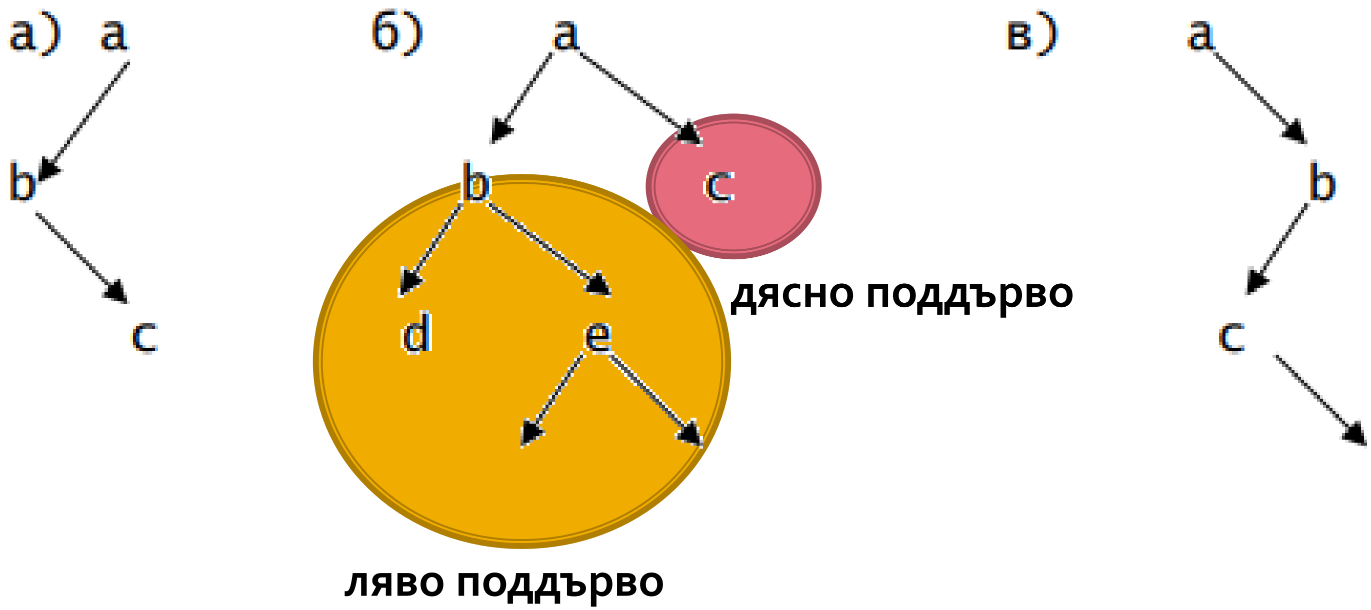
Двоично дърво

Двоично дърво от тип Т е рекурсивна структура от данни, която е или празна или е образувана от:

- Данна от тип Т, наречена **корен** на двоичното дърво;
- Двоично дърво от тип Т, наречено **ляво поддърво** на двоичното дърво;
- Двоично дърво от тип Т, наречено **дясно поддърво** на двоичното дърво.

Двоично дърво

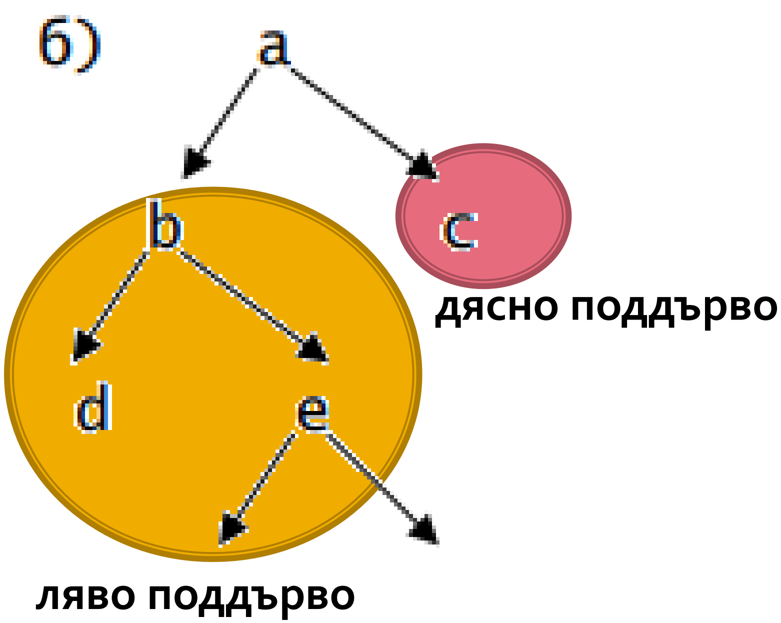
Пример:



Двоично дърво

Множеството на **върховете (възлите**) на едно двоично дърво се определя рекурсивно:

- Празното двоично дърво няма върхове.
- Върховете на непразно дърво са неговият корен и върховете на двете му поддървета.




```
// a, b, d, e, c
```

Двоично дърво

- **Листа** – върховете с две празни поддървета.
- **Вътрешни върхове** – върховете, различни от корена и листата.
- **Ляв наследник** на един връх – коренът на лявото му поддърво (ако то е непразно).
- **Десен наследник** на един връх – коренът на дясното му поддърво (ако то е непразно).
- Ако a е наследник на b (ляв или десен), казваме, че b е **родител** (**баща**) на a.

Двоично дърво

- **Ниво** – коренът на дървото има ниво 1 (или 0). Ако един връх има ниво i, то неговите наследници имат ниво i+1.
- **Височина (дълбочина)** – максималното ниво на едно дърво.

Двоично дърво

- **Достъп до връх** – възможен е пряк достъп до корена и непряк достъп до останалите върхове.
- **Операции** – <u>възможни са</u> операциите добавяне и премахване на върхове на произволно място в двоичното дърво, <u>но резултатът трябва отново да е двоично дърво</u> от същия тип. Как ще се извършва добавянето и изтриването?
- **Обхождане** – това е метод, позволяващ да се осъществи достъп до всеки връх на дървото един единствен път.

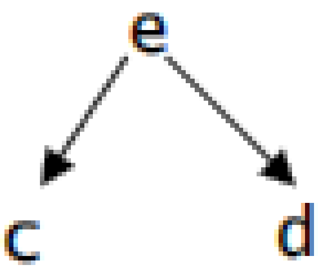
Двоично дърво

Обхождането е рекурсивна процедура, която се осъществява чрез изпълнение на следните три действия, в някакъв фиксиран ред:

- обхождане на корена
- обхождане на лявото поддърво
- обхождане на дясното поддърво

Двоично дърво

- **Смесеното обхождане (ЛКД)** – c, e, d
- **Низходящото обхождане (КЛД)** – e, c, d
- **Възходящо обхождане (ЛДК)** – c, d, e



Съществуват още три типа обхождания.

- **КДЛ** – e, d, c
- **ДКЛ** – d, e, c
- **ДЛК** – d, c, e

Двоично дърво

Физическо представяне на двоично дърво:

- Свързано
- Последователно

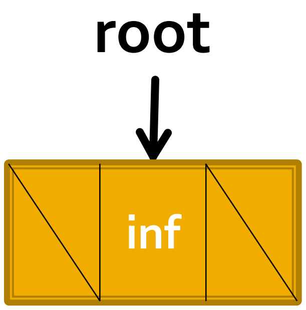
Двоично дърво

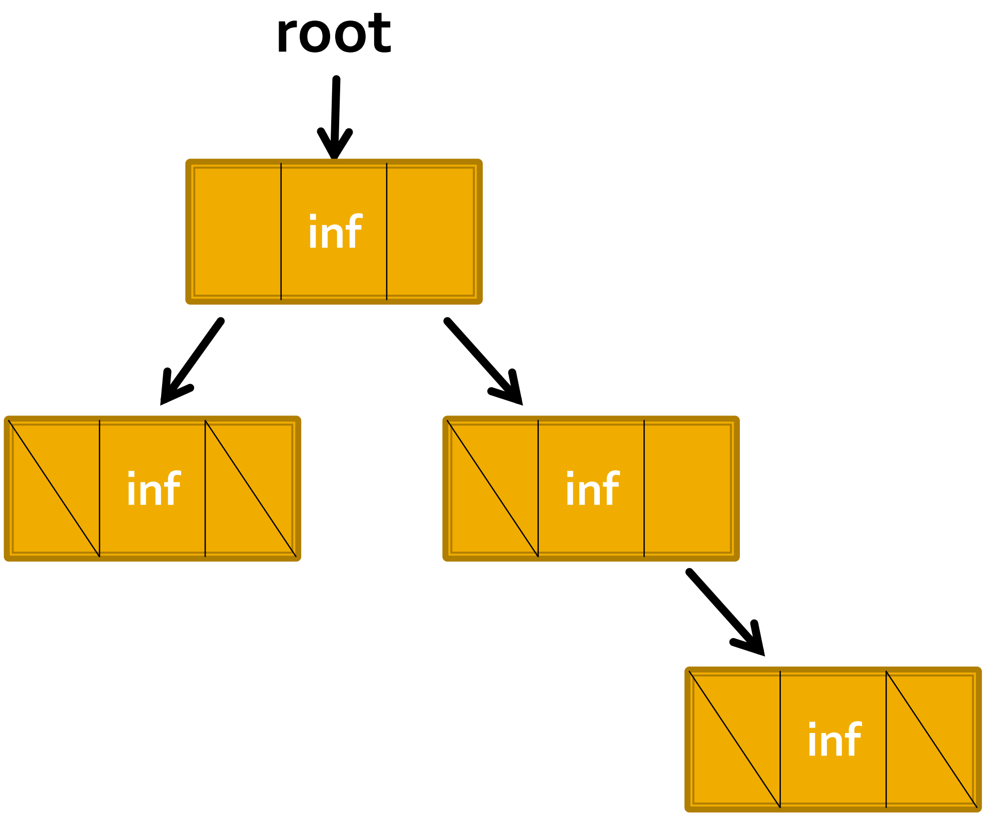
Физическо представяне на двоично дърво:

- Свързано
- Последователно
  - Верижно
  - Списък на бащите

Двоично дърво

Свързано представяне





Двоично дърво

Верижно представяне

Използват се три масива – **`a[N]`**, **`b[N]`** и **`c[N]`**.

** N е броят на върховете в дървото*

Върховете са номерирани от 0 до N-1.

- `a[i]` – стойността на i-тия връх на дървото.
- `b[i]` – индексът на левия наследник на i-тия връх (-*1, ако той няма ляв наследник*).
- `c[i]` – индексът на десния наследник на i-тия връх (-*1, ако той няма десен наследник*).
- Индексът на корена – пази се отделно.

Двоично дърво

Чрез списък на бащите

Представя се с един масив `p[N]`.

* *N е броят на върховете в дървото*

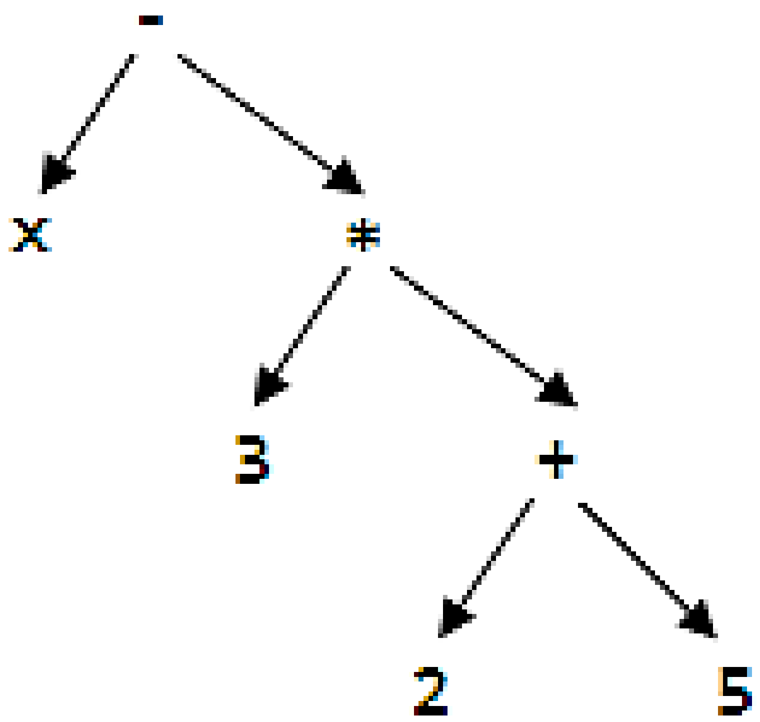
Върховете са номерирани от 0 до N-1

- `p[i]` - единственият баща на i-тия връх на дървото (-*1, ако този връх е коренът*).

https://www.geeksforgeeks.org/construct-a-binary-tree-from-parent-array-representation/

Двоично дърво

Представяне на изрази
x – 3 * (2 + 5)



Двоично дърво

Представяне на изрази

- Възходящия обход (ЛДК) – обратен полски запис

Пример:

x 3 2 5 + * -

- Смесеният обход (ЛКД) – инфиксен запис на аритметичния израз *(без скобите)*

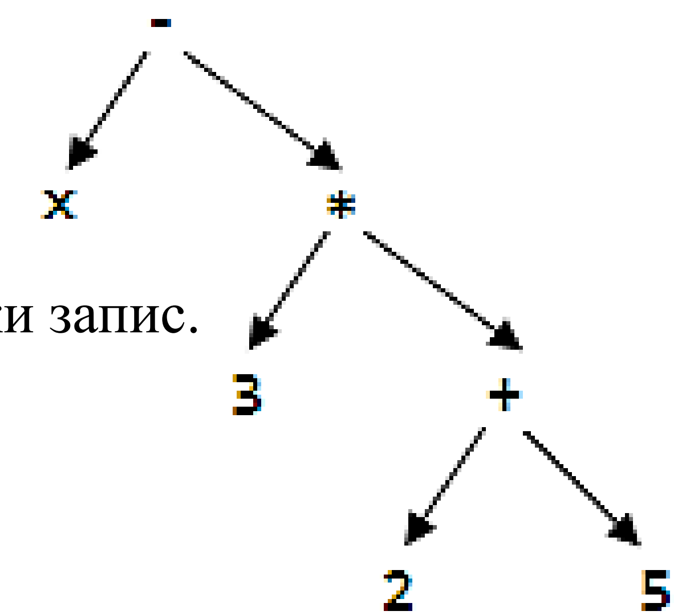
Пример:

x – 3 * 2 + 5

- Низходящият обход (КЛД) – прав полски запис.

Пример:

– x * 3 + 2 5



Сложност на операциите

- Средна сложност на операциите добавяне и търсене – `O(log N)` и `O(N)`

Следва продължение...