

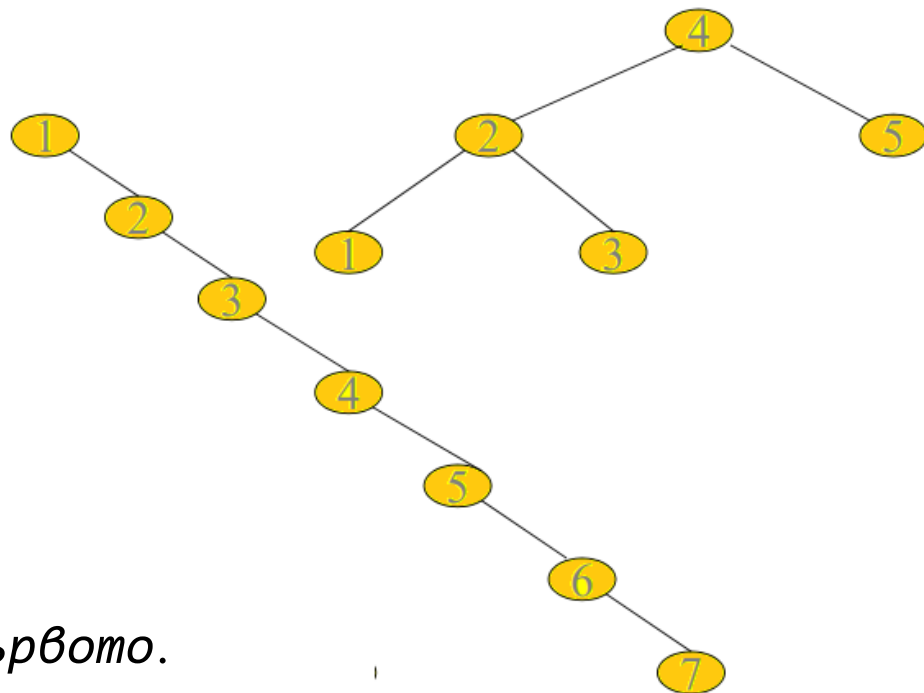
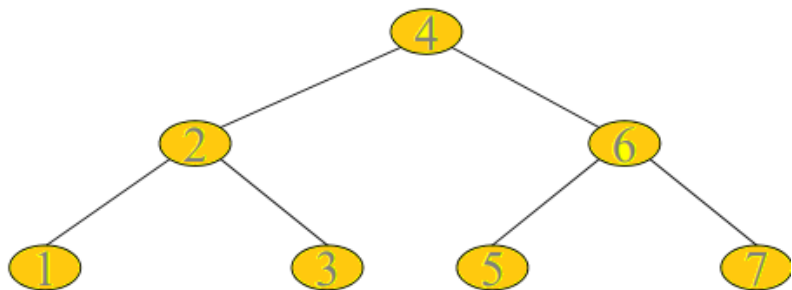
Балансирано и идеално балансирано дърво

доц. д-р. Нора Ангелова

Двоично наредено дърво

Двоични наредени дървета

- Средна сложност на операциите добавяне и търсене - $O(\log N)$.
- Добавяне на елементите подредени по големина - $O(N)$.



* N – брой на върховете в дървото.

Идеално балансирано дърво

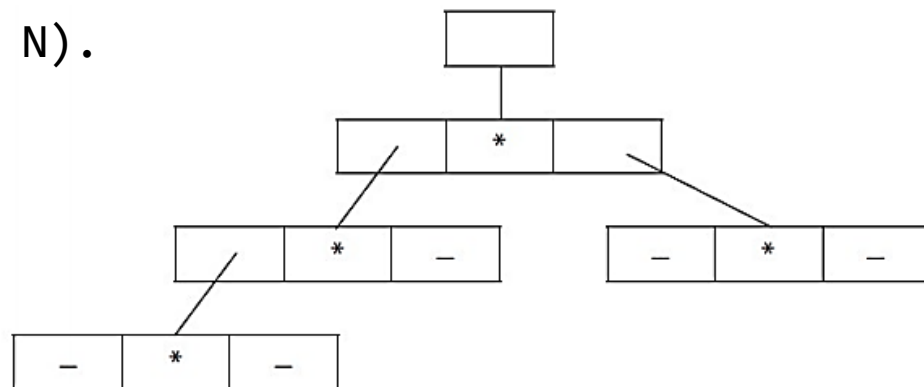
Дефиниция

Двоично наредено дърво се нарича идеално (перфектно) балансирано, ако всеки негов връх има ляво и дясно поддърво, в които **броят на възлите** се различава най-много с 1.

Двоично наредено дърво със следните свойства:

- Броят на възлите в лявото и дясното поддърво се различава най-много с 1.
- Лявото и дясното поддървета са идеално балансирани двоично наредени дървета.

Сложност на операциите - $O(\log N)$.



* N – брой на върховете в дървото.

Балансирано дърво (AVL)

Дефиниция

Двоично наредено дърво се нарича двоично дърво с балансирана височина или само балансирано дърво, ако за всеки негов връх **височините** (дълбочините) на лявото и дясното му поддървета се различават най-много с 1.

Двоично наредено дърво със следните свойства:

- Височините на лявото и дясното поддърво се различават най-много с 1.
- Лявото и дясното поддървета са балансирани двоично наредени дървета.

Сложност на операциите - $O(\log N)$.

* N – брой на върховете в дървото.

Балансирано и идеално балансирано дърво

Свойства:

- Всяко идеално балансирано дърво е балансирано дърво, но обратното не е вярно.
- Алгоритмите за добавяне и премахване на връх от двоично наредено дърво не запазват балансираността и добрите сложности.
- Съществуват алгоритми, които запазват балансираността.
- Съществува прост алгоритъм за създаване на идеално балансирано двоично наредено дърво при определени ограничения.

Балансирано и идеално балансирано дърво

Алгоритъм за създаване на идеално балансирано дърво

- Елементите, които ще се включват към празното двоично наредено дърво се подават в нарастващ ред.
- Предварително е известен броят на върховете на дървото.

n - брой на върховете в дървото.

$$n = n_{\text{Left}} + n_{\text{Right}} + 1 \ \&\& \ |n_{\text{Left}} - n_{\text{Right}}| \leq 1$$

Балансирано и идеално балансирано дърво

Алгоритъм за създаване на идеално балансирано дърво

n - брой на върховете в дървото.

$$n = nLeft + nRight + 1 \ \&\& \ |nLeft - nRight| \leq 1$$

- Конструира ляво поддърво с $nLeft$ върха.
- Създава корен на дървото.
- Конструира дясно поддърво с $nRight$ върха.

Балансирано дърво

Физическо представяне

Четворна кутия – добавя се още един параметър, означаващ коефициент на балансираност, задаващ разликата на височините на дясното и лявото поддърво на съответния връх.

Балансирано дърво

- **Ротации** - операции, които пренареждат част от елементите на дървото при добавяне или при премахване на елемент от него, за да се избегне дисбалансът му.
- Ротациите зависят от реализацията на конкретната структура от данни. Примери за такива структури:
 - **червено-черно** дърво
 - **AVL** - дърво
 - **AA** - дърво и др.

Балансирано и идеално балансирано дърво

AVL дърво

<https://www.cs.usfca.edu/~galles/visualization/AVLtree.html>

- **Търсене** – осъществява се по същия начин както при обикновено небалансирано двоично наредено дърво.

Най-лош сценарий:

Трябва да се обходи от корена до най-далечното листо.

Време: пропорционално на височината на дървото – $O(\log n)$.

Балансирано дърво

AVL дърво

- **Обхождане** – осъществява се по същия начин както при обикновено небалансирано двоично наредено дърво.

Обхождат се всички върхове – $O(n)$.

Балансирано дърво

AVL дърво

Добавяне на връх

Операцията включване на елемент в балансирано двоично наредено дърво се осъществява по аналогичен начин на включването на елемент в двоично наредено дърво. Разликата е, че при всяко включване, което променя балансираността на дървото трябва да се изпълнят едно или две завъртания с цел възстановяване на балансираността.

Балансирано дърво

AVL дърво

Добавяне на връх:

- Върхът се добавя като листо.
- Повторно обхождане (**retracing**) - проверка за съответствие с инвариантите на AVL дърво. Започва се от родителя на добавения връх и се стига до корена.

Реализира се чрез пресмятане на баланс фактор за всеки връх, който се дефинира като разликата във височините на лявото и дясното поддървета.

`balanceFactor` – цяло число в интервала `[-1 1]`.

Балансирано дърво

AVL дърво

Добавяне на връх:

- Връх с `balanceFactor` $\in [-1 \ 1]$ - поддървото е балансирано и не е необходима ротация.
- Връх с `balanceFactor` $\in [-2 \ 2]$ – поддървото е небалансирано и е необходима ротация.

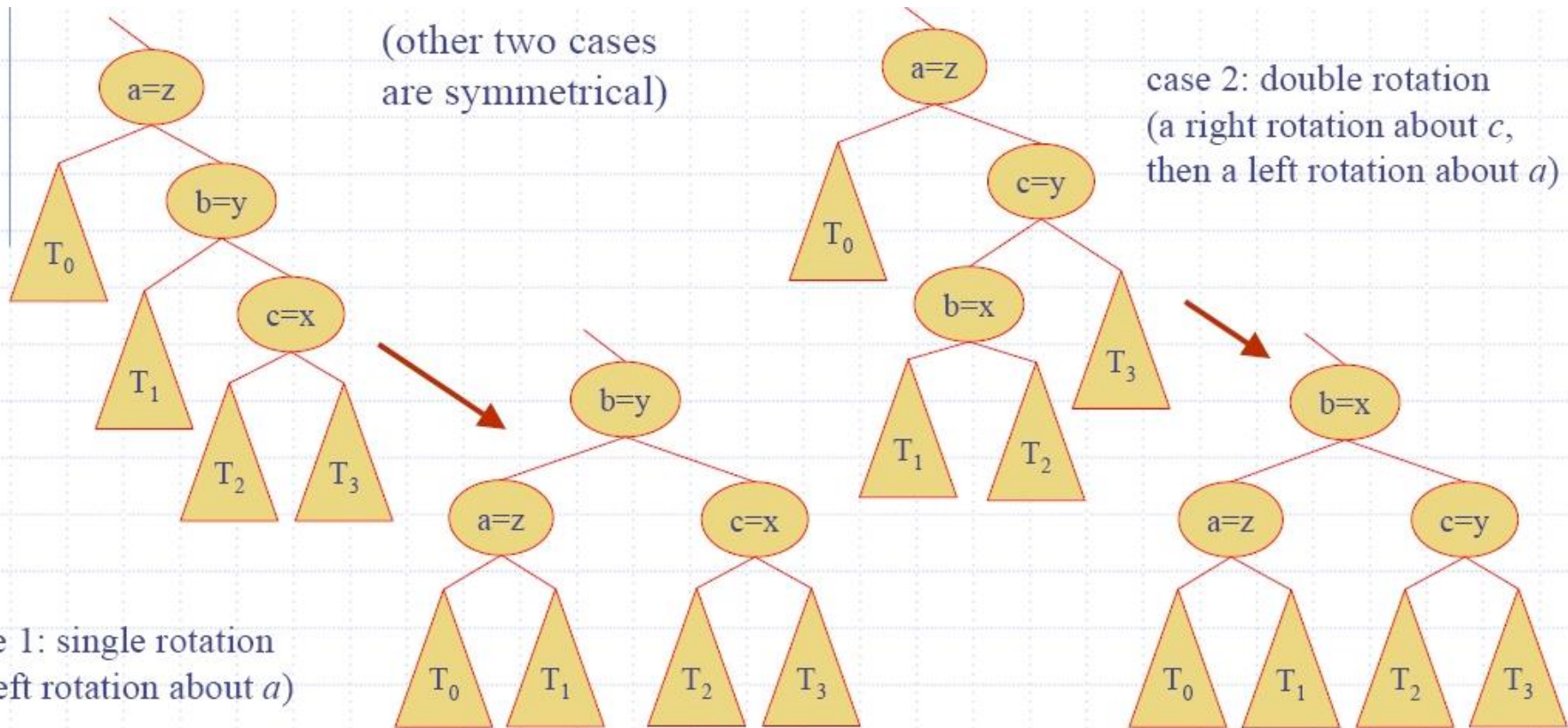
Балансирано дърво

AVL дърво

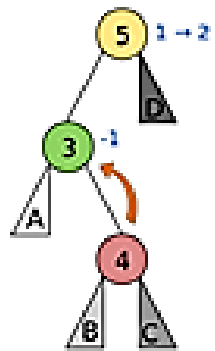
Ротационни функции:

- Дясна ротация.
- Лева ротация.
- Двойна ляво-дясна ротация (състои се от лява ротация, последвана от дясна).
- Двойна дясно-лява ротация (състои се от дясна ротация, последвана от лява).

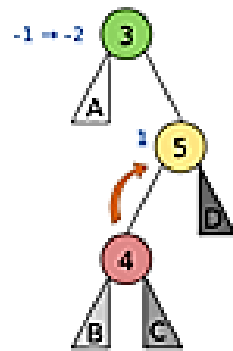
Балансирано дърво



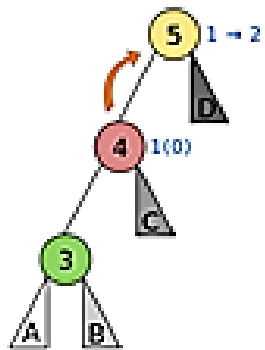
ліво-ліва ротация



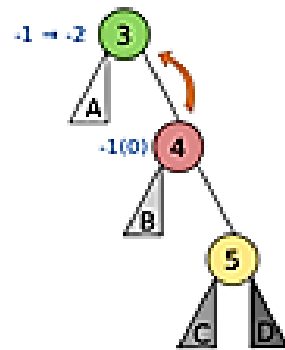
ліво-ліва ротация



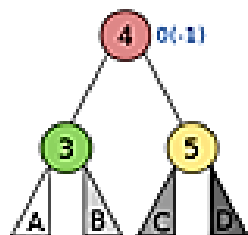
двойна ліва



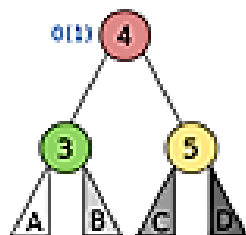
двойна ліва ротация

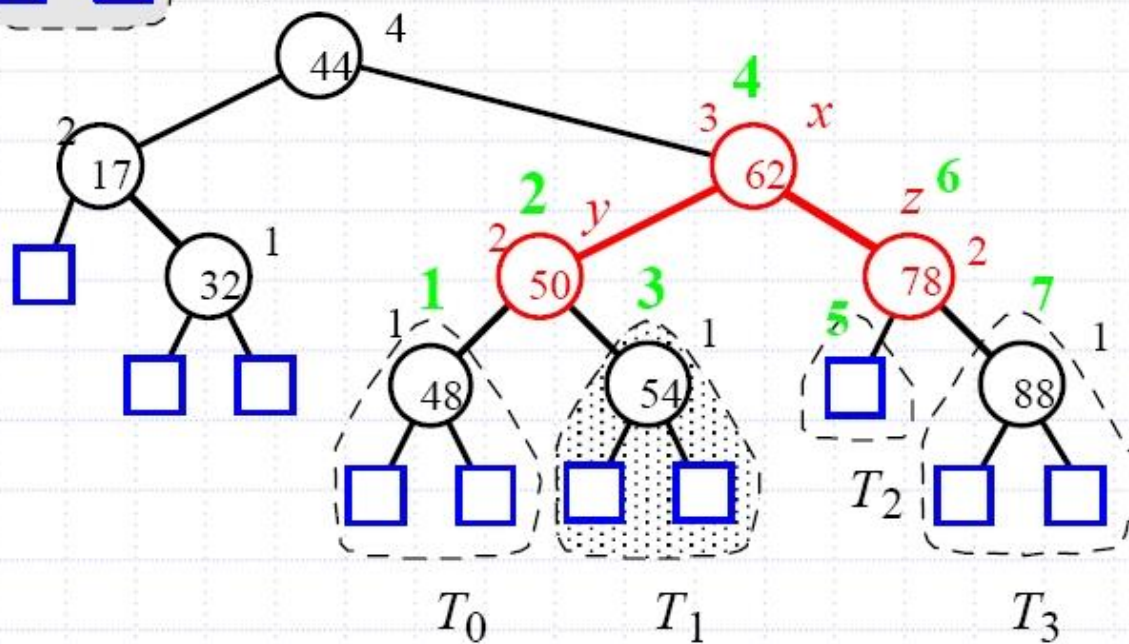
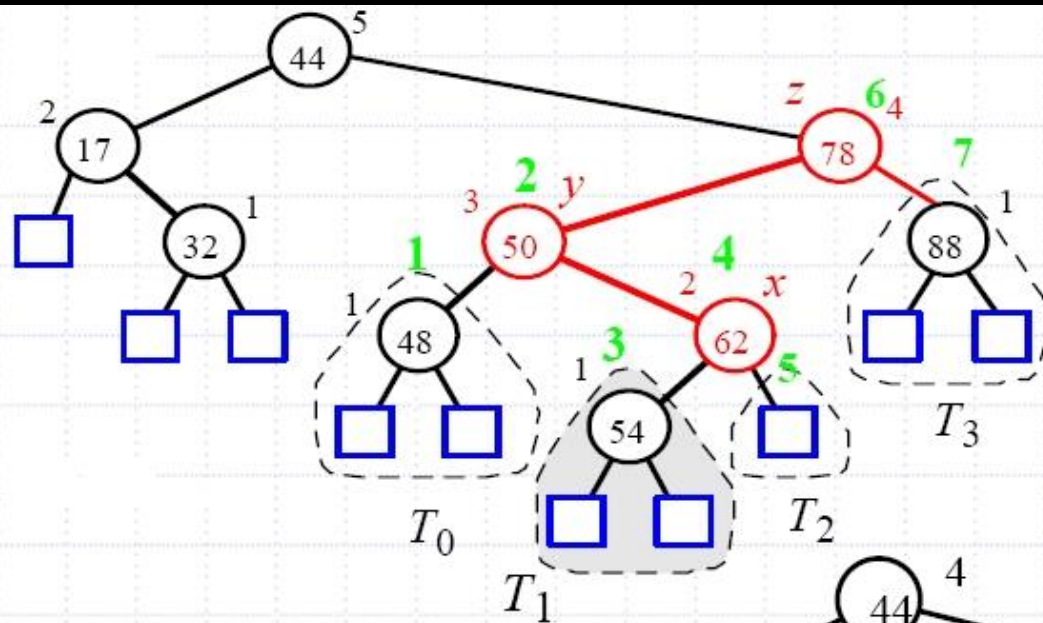


балансировано



балансировано





Балансирано дърво

AVL дърво

Изтриване на връх:

- Изтриване на връх от дървото.
- Повторно обхождане (**retracing**) - проверка за съответствие с инвариантите на AVL дърво. Започва се от родителя на изтрития връх и се стига до корена.

Следва продължение...