Балансирани Дървета

Лекция 8 по СДА, Софтуерно Инженерство
Зимен семестър 2018-2019г
д-р Милен Чечев

От предишните лекции

Двоичното дърво за търсене(BST) в най-лошия случай е със линейна сложност за търсене и добавяне на елемент

implementation	guarantee		average case	
	search	insert	search hit	insert
sequential search (unordered list)	n	n	n	n
binary search (ordered array)	$\log n$	n	$\log n$	n
BST	n	n	$\log n$	$\log n$

Каква структура да използваме ако само ще търсим без да променяме числата в структурата?

- Масив
- Как?
  - 1. Сортираме го
  - 2. Търсим със сложност O(log(n))

Каква структура да използваме ако ще търсим, но също така ще добавяме и изваждаме числа?

Балансирано дърво.

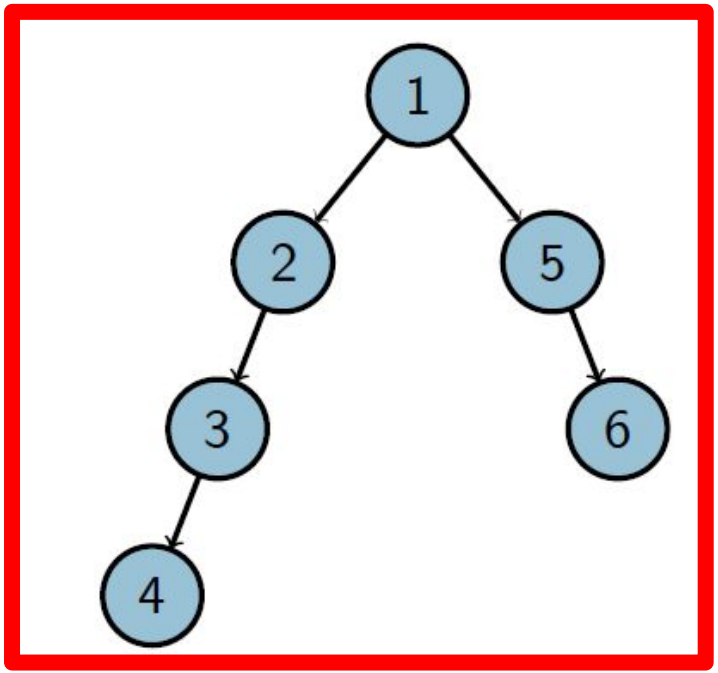
Дефиниция: Дърво, в което за всеки възел имаме свойството, че височината на лявото му поддърво се различава от височината на дясното поддърво с максимум единица.

Сложности в най-лошият случай:

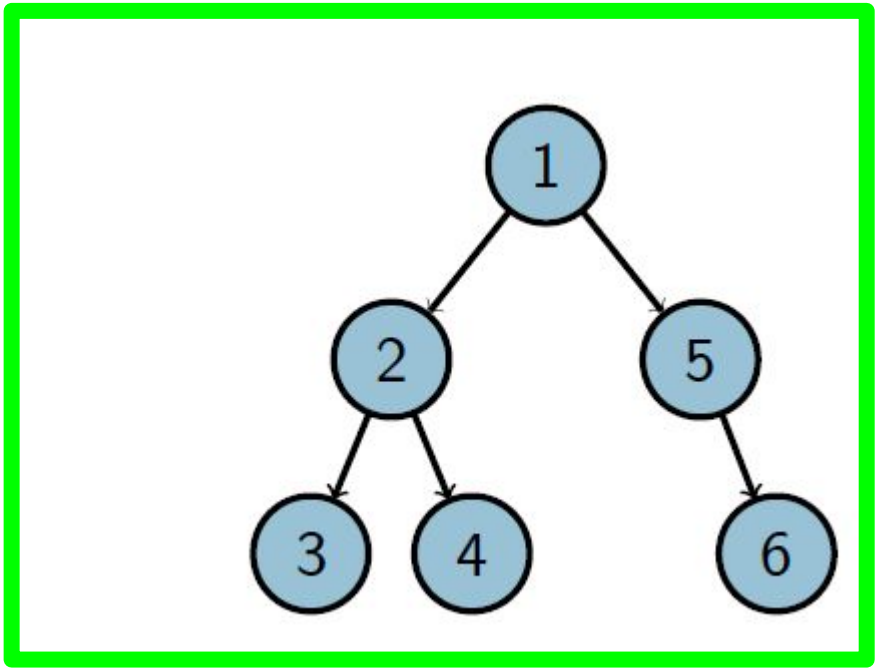
- Търсене O(log(n))
- Добавяне на елемент O(log(n))
- Изтриване на елемент O(log(n))

Това балансирано дърво ли е ?

НЕ



ДА



Видове балансирани дървета

- 2-3 дърво
- AVL дърво
- Red-Black дърво
- Splay дърво
- Treap

AVL дърво

- Предложено от Адельсон-Велский и Ландис през 1962 г.
- Основна идея:

  Всяко поддърво T = (X, L, R) поддържа коефициент на баланс:

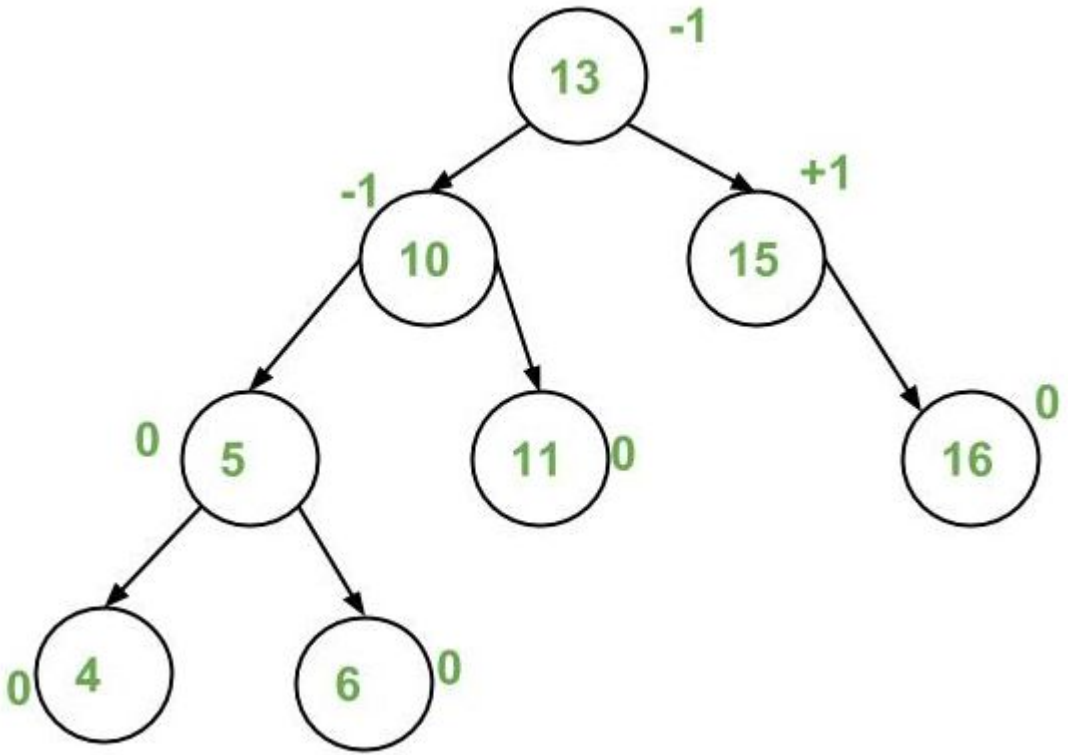
  $$b(T) = h(R) - h(L)$$

  Като се изисква винаги коефициента на баланс да е [-1,0,1]

Как AVL дървото запазва баланса си?

- Дървото преди и след всяка операция задължително се намира в състояние, че за всеки възел X имаме $b(X) \in (-1,0,1)$.
- Операциите добавяне и премахване първоначално се извършват стандартно като за BST и оттам понеже имаме промяна на броя на възлите може да имаме и промяна на балансиращият индекс за бащите на добавеният или премахнат възел.
  - Промяната може да направи стойности на b от -2 или 2.
- След стандартната процедура за добавяне или премахване ако е необходимо се прилагат ротации за балансиране на дървото.

Илюстративен пример

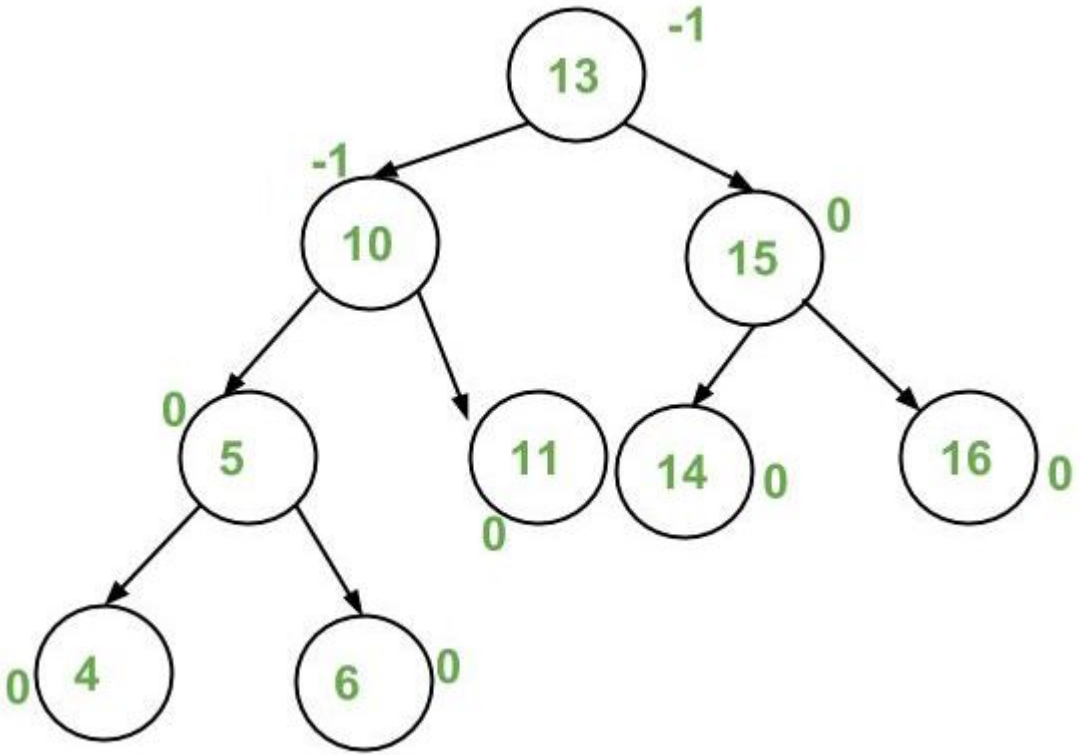


Ако искаме да добавим 14

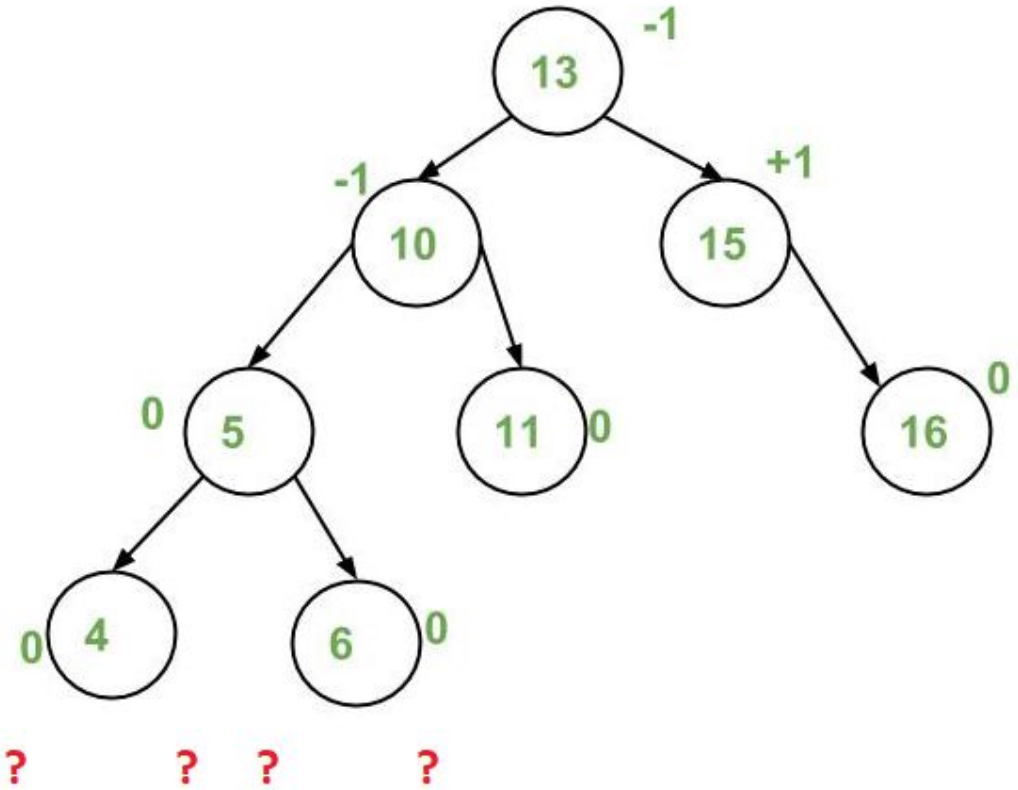
Нямаме никаква разлика от

Добавяне в BST

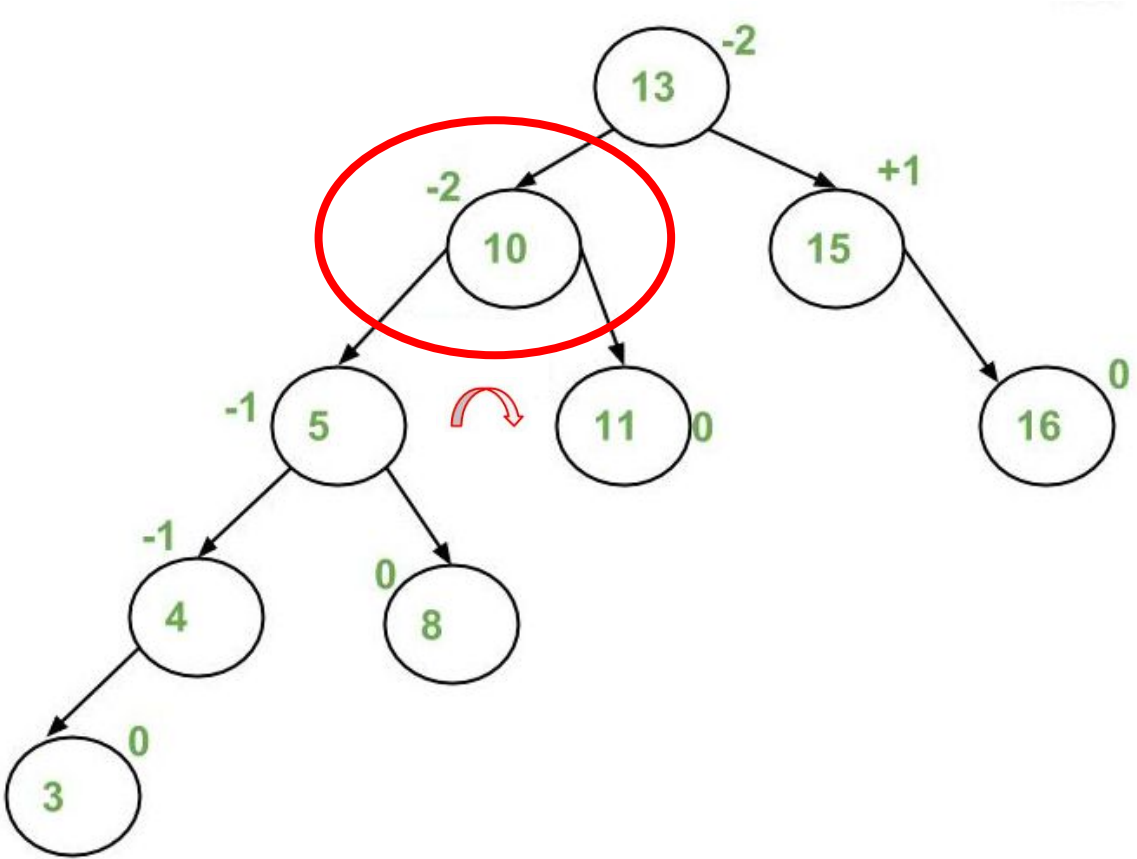
Резултат:



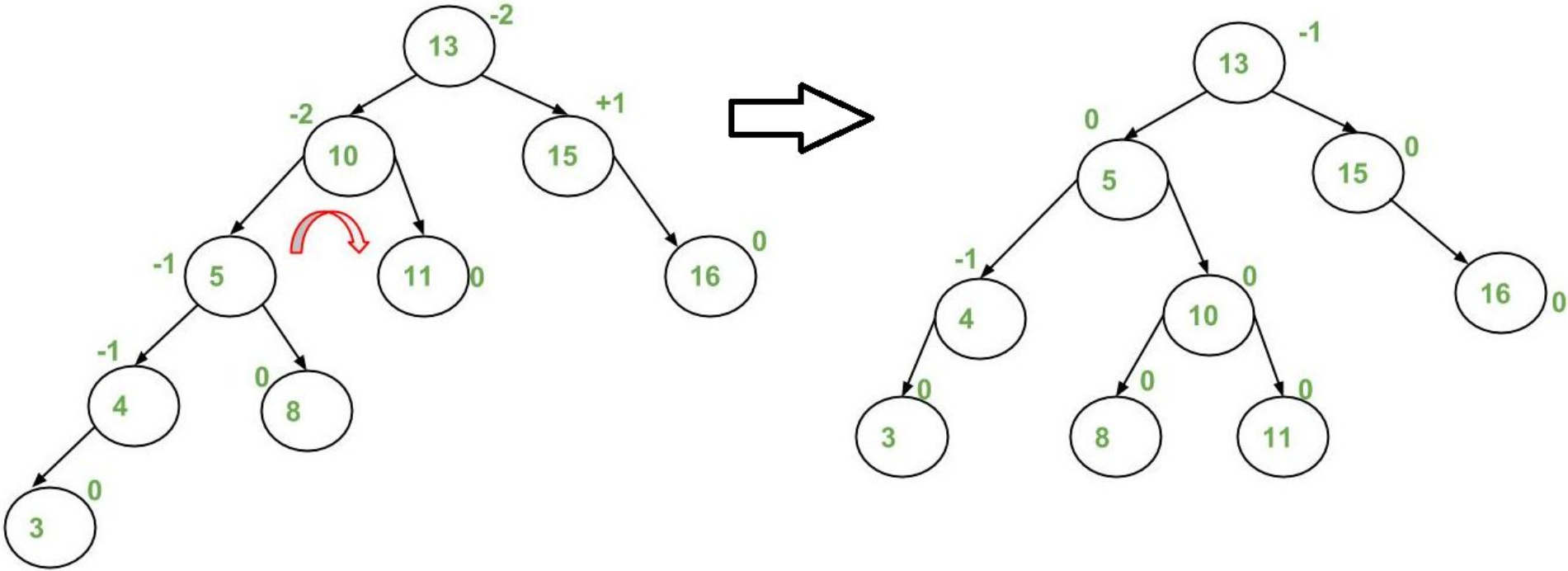
Какво ще стане обаче ако ...?



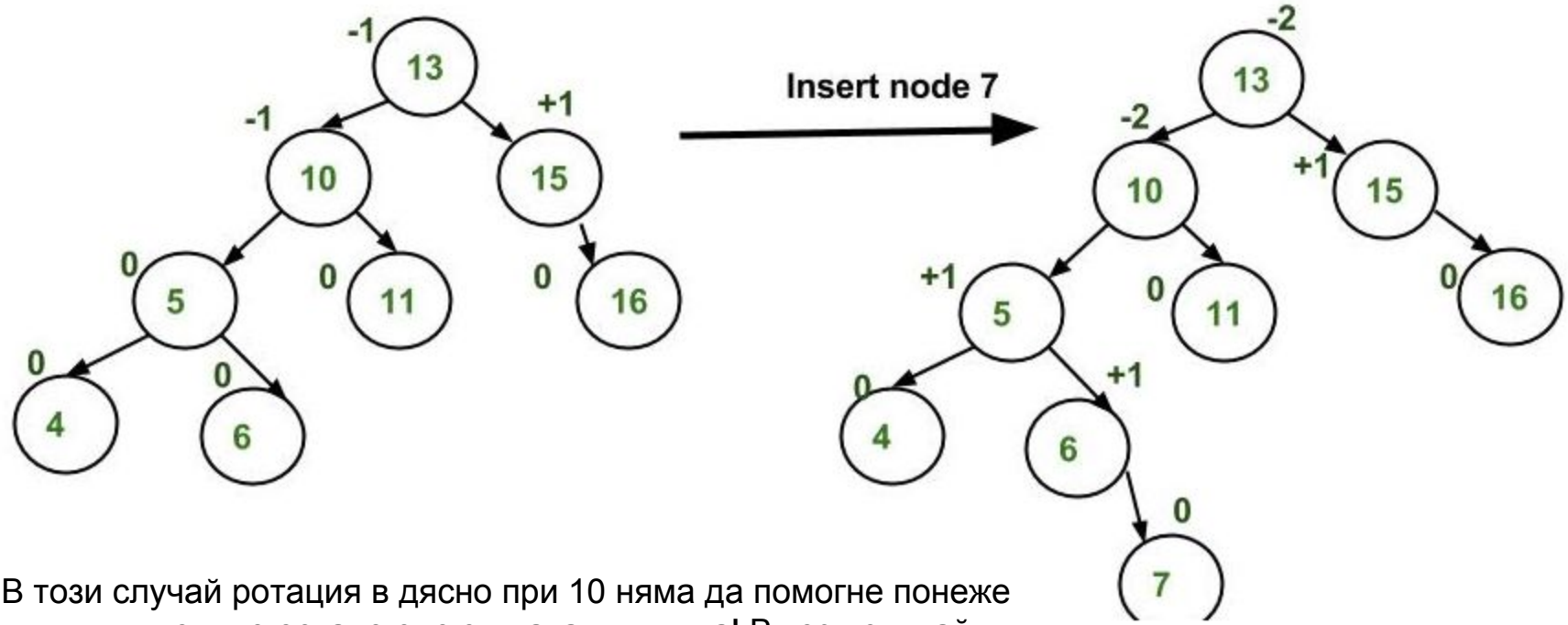
След добавянето трябва балансиране!



Резултат:

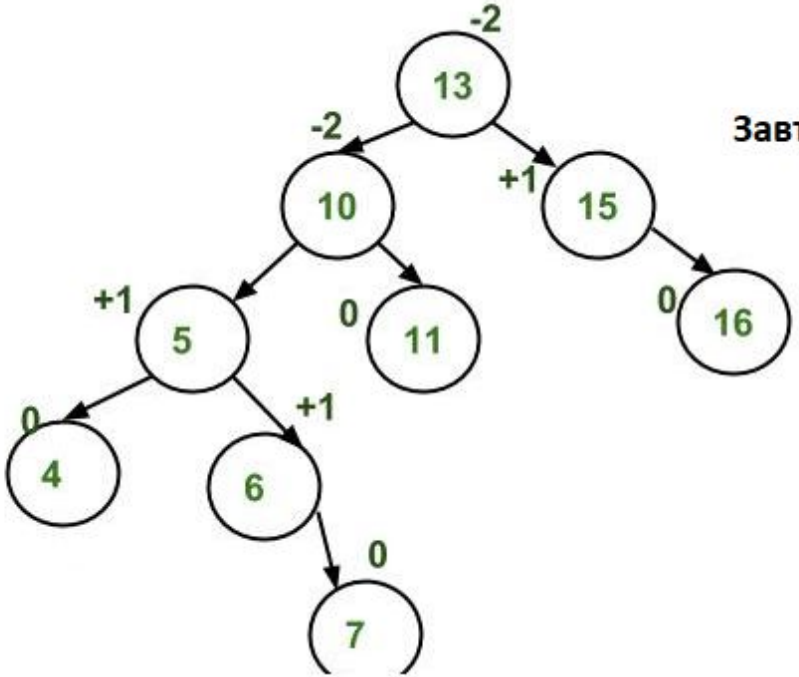


А ако новия възел е първо в ляво после в дясно ?

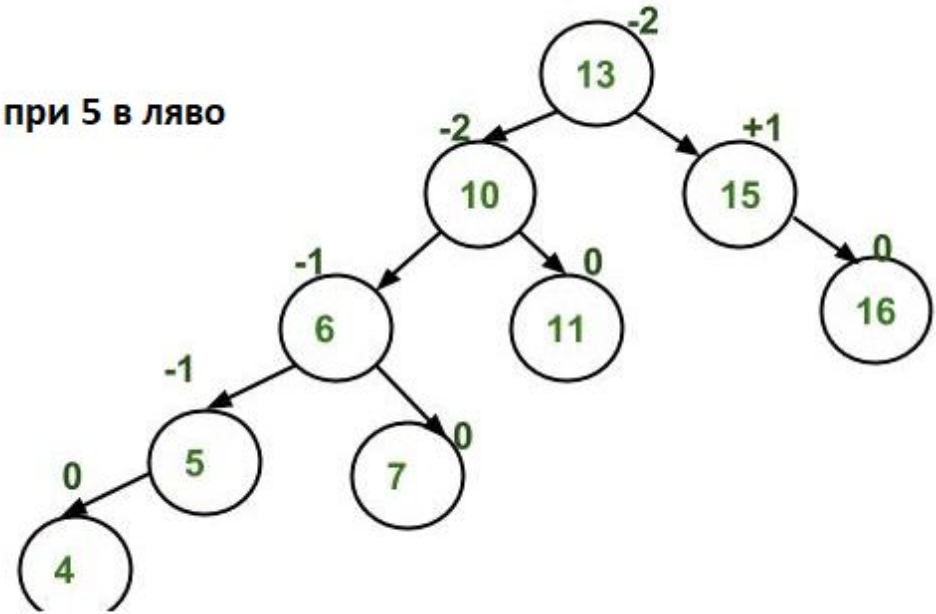


В този случай ротация в дясно при 10 няма да помогне понеже дългият клон ще остане със същата дължина! В този случай въртим първо в ляво при 5 и чак след това в дясно при 10

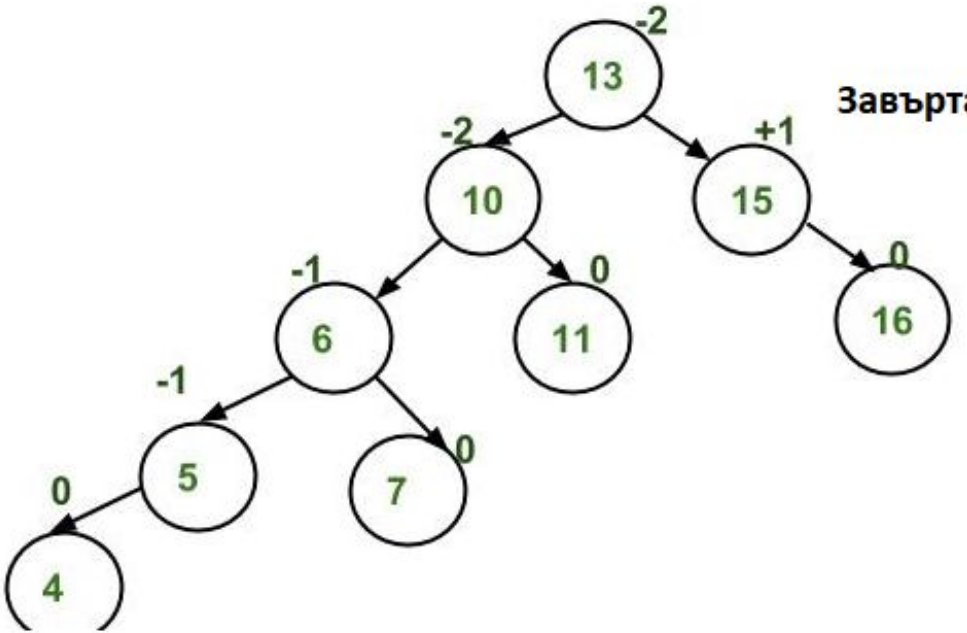
Първо завъртане - в ляво преди възела с лош индекс



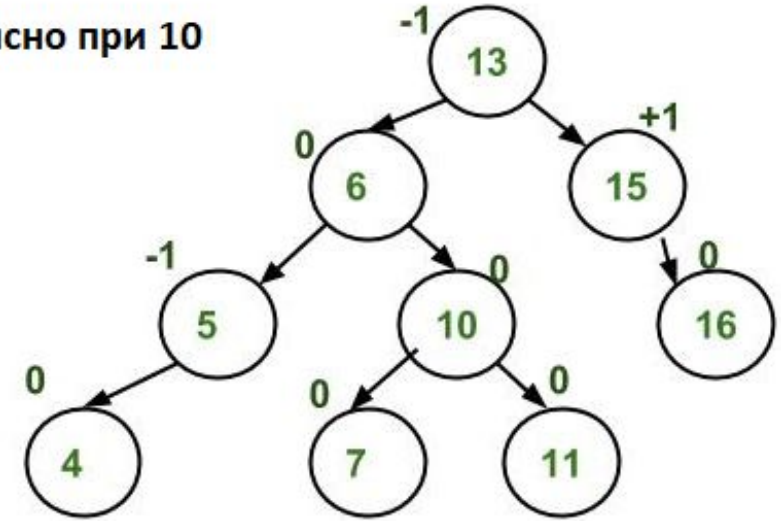
Завъртане при 5 в ляво



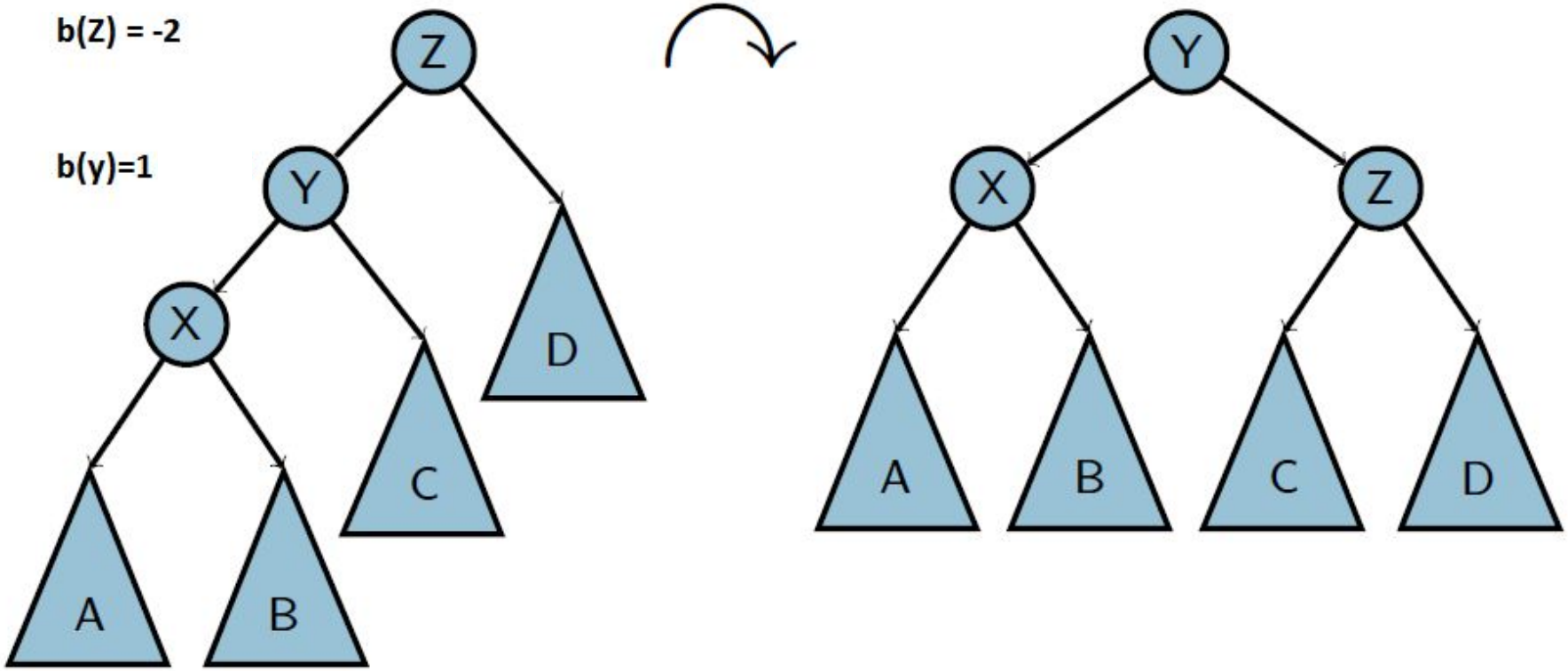
Второ завъртане във възела с лош индекс



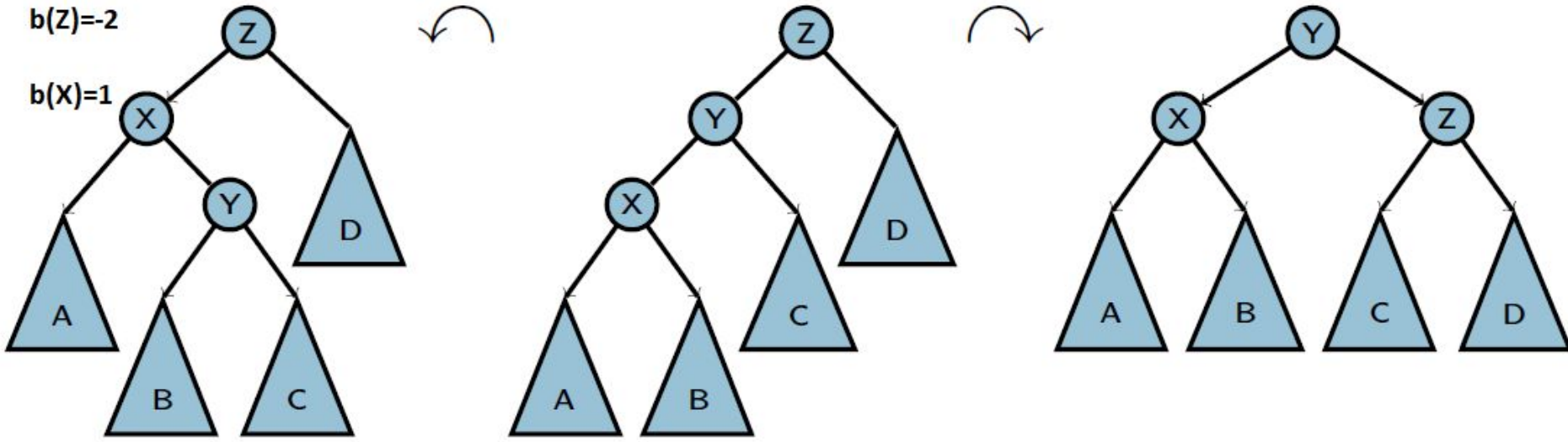
Завъртане в дясно при 10



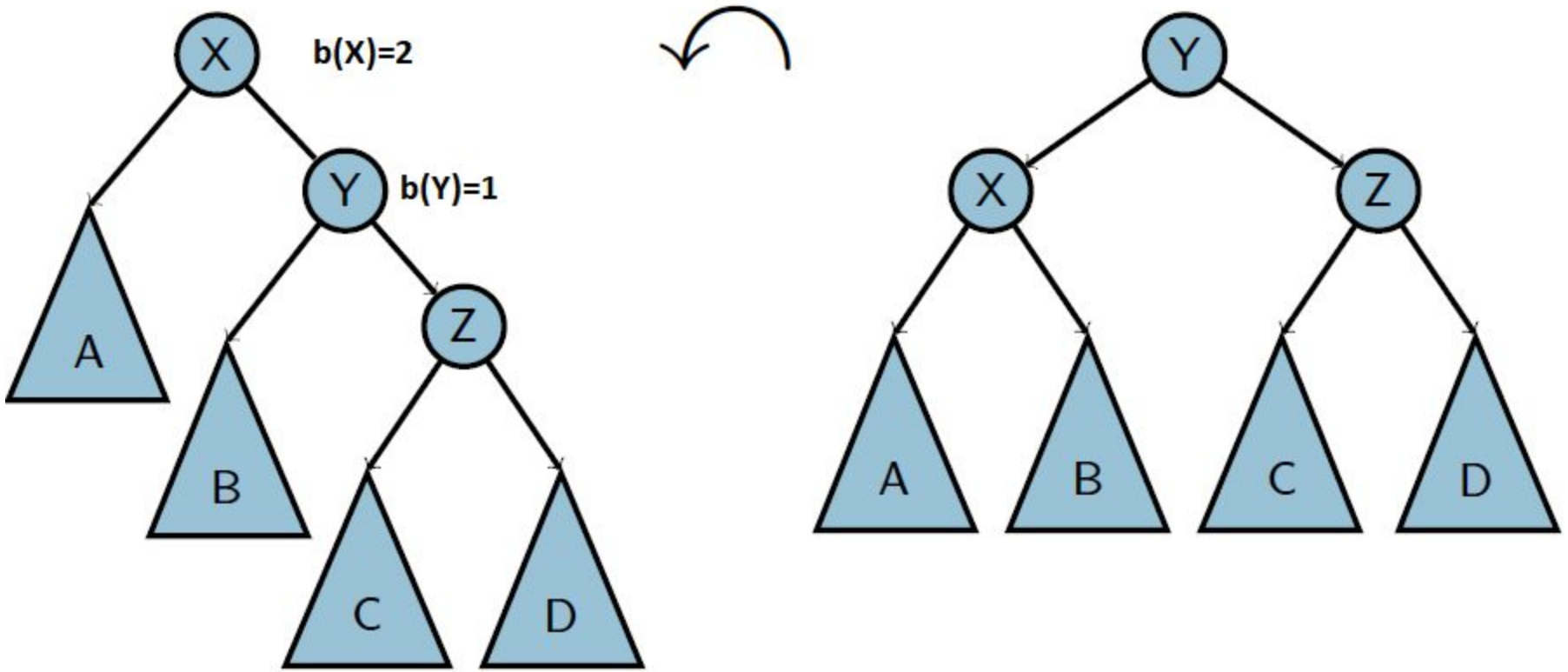
Обобщение. Балансиране със завъртане в дясно



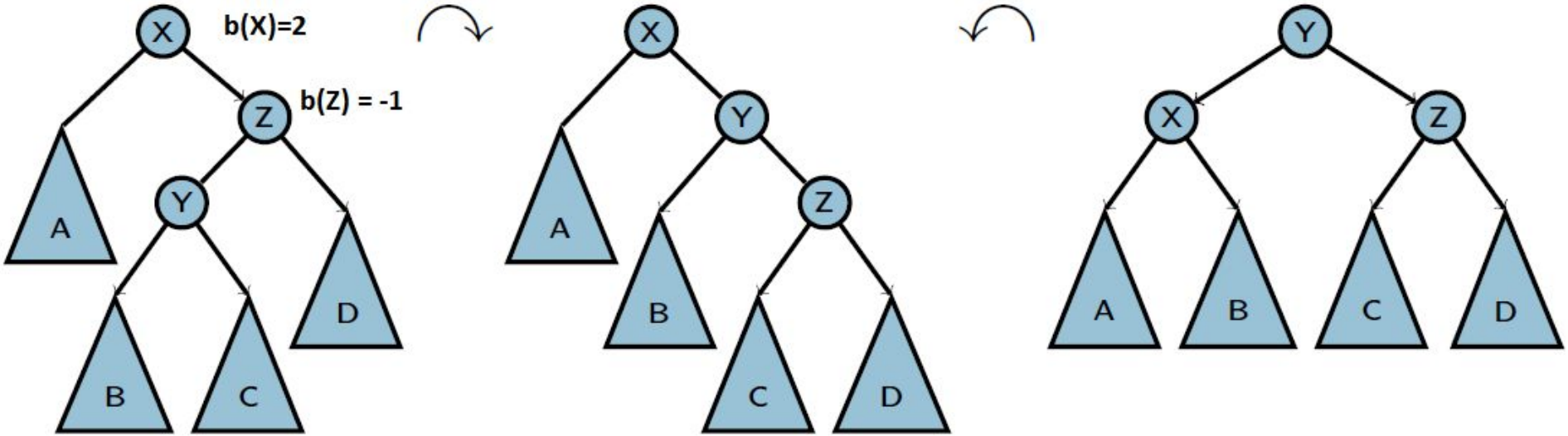
Балансиране със завъртане първо в ляво после в дясно



Балансиране със завъртане в ляво



Балансиране със завъртане първо в дясно после в ляво



Балансиране при включване и изключване

Балансиране при включване

- При дъното на включването височината винаги се увеличава с 1
- Ако височината на по-ниското дете се увеличи, то височината на родителя не се променя
- При балансиране винаги компенсираме за увеличената височина на детето

Балансиране при изключване

- При дъното на изключването дъното височината винаги се намалява с 1
- Ако височината на по-високото дете се намали, то височината на родителя не се променя
- Ако след балансиране b(T) != 0, значи сме компенсирали за намалената височина на детето
- Ако след балансиране b(T) = 0, значи височината се е намалила

Какво следва?

- Домашно за реализация и използване на балансирани дървета
- Сравнение на балансирани дървета с хеш таблица(лекция 9)