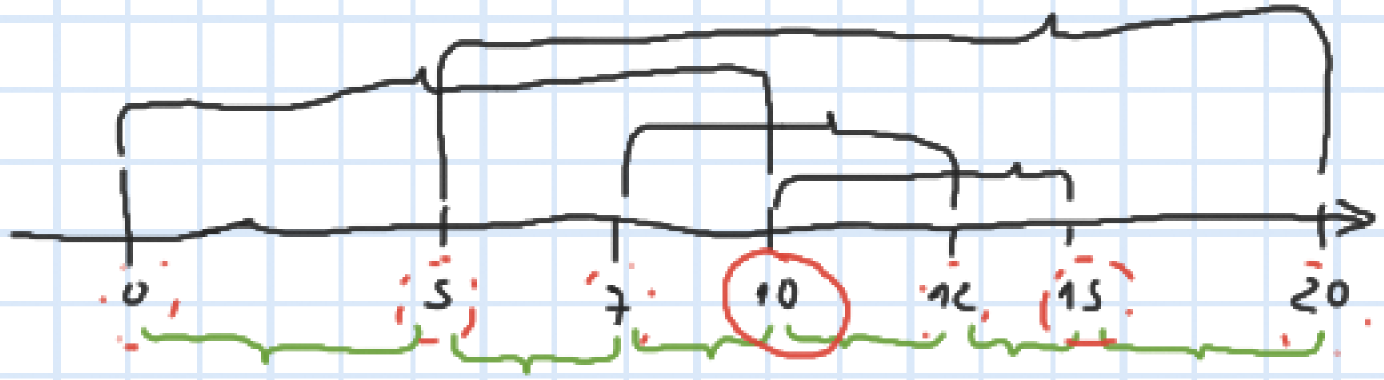
Algorytmy i Struktury Danych
Wykład 15

Drzewa przedziałowe — interesuje nas struktura przechowująca przedziały i pozwalająca odczytać listę przedziałów zawierających dany punkt

Przykład

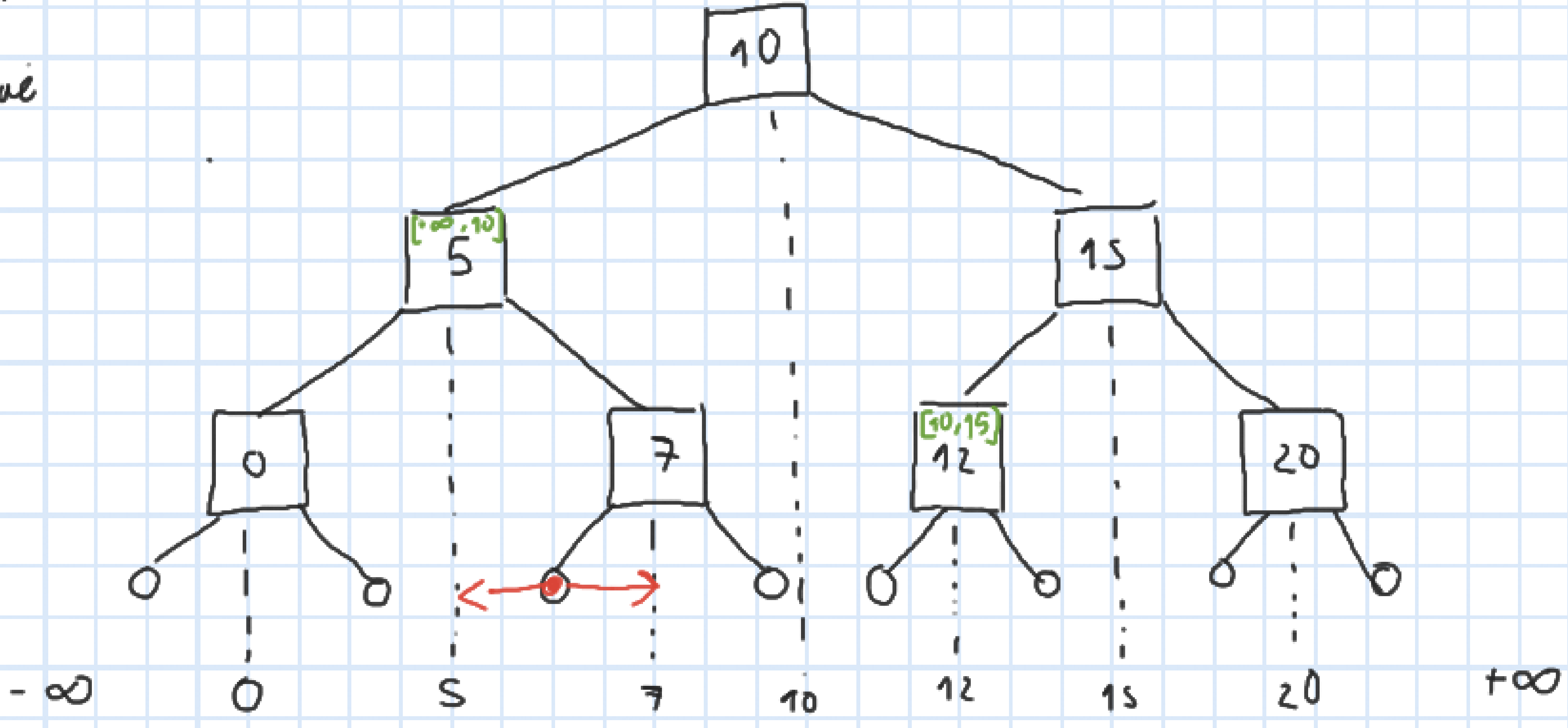
Rozważmy przedziały

$[0,10]$, $[5,20]$, $[7,12]$, $[10,15]$



przedziały bazowe

Konstruujemy drzewo BST zawierające punkty graniczne przedziałów bazowych

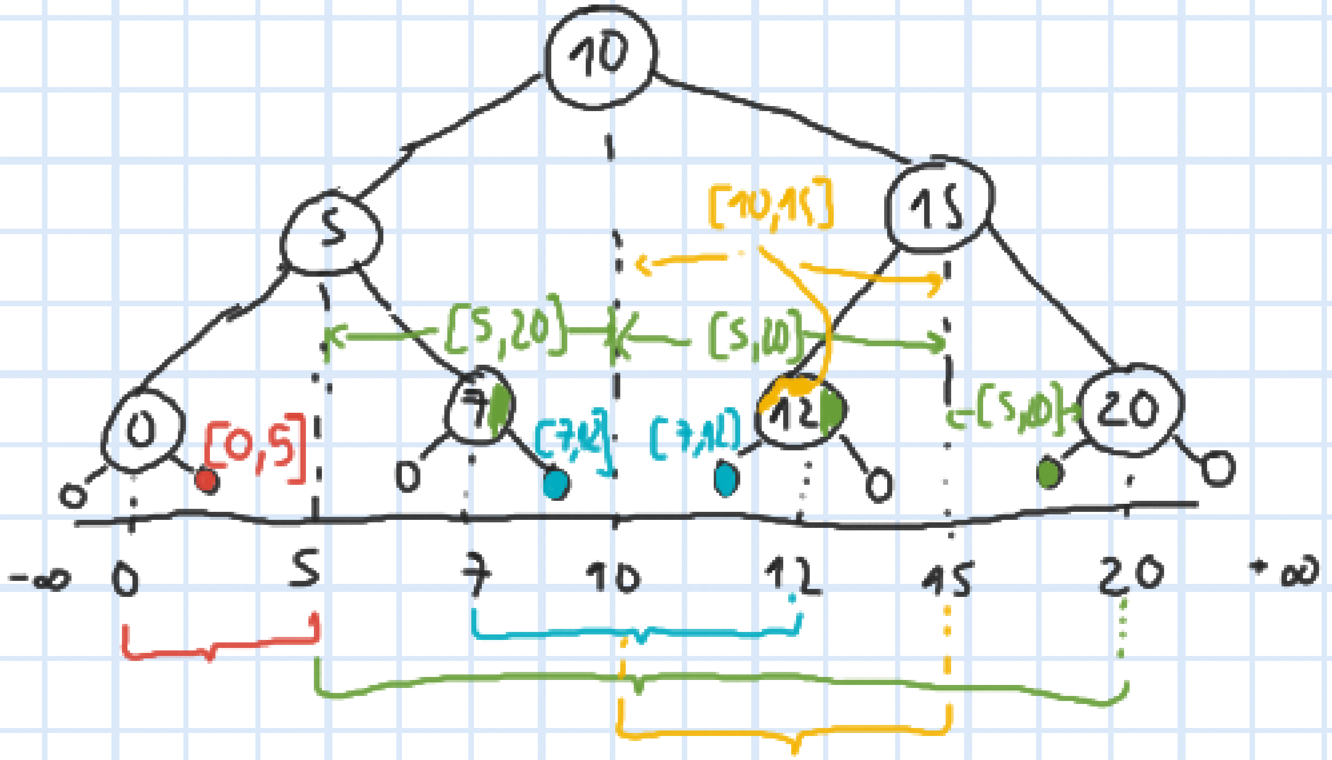


Każdy liść odpowiada za dany przedział bazowy

Każdy węzeł odpowiada za sumę przedziałów bazowych w jego poddrzewach

$\text{span}(10) = [-\infty, \infty]$, $\text{span}(7) = [5,10]$, $\text{span}(15) = [10, \infty]$

Przedział przechowujemy w tych węzłach, których span całkowicie się w tym przedziale zawiera (a nie jest to prawda dla rodzica węzła)



<u>Złożoność pamięciowa</u> – każdy przedział na danym poziomie drzewa występuje najwyżej 2 razy ⇒ n przedziałów daje złożoność pam. $O(n \log n)$

Ustawianie przedziału

Zaczynając od korzenia rozważamy węzły

- Jeśli przedział "pasuje idealnie" (jest równy span'owi węzła) to ustawiamy go.
- Jeśli nie to:
  - przecięcie przedziału ze span'em lewego dziecka ustawiamy po lewej stronie (rek.)
  - przecięcie przedziału ze span'em prawego dziecka ustawiamy po prawej

ustawiam w danym poddrzewie o ile przecięcie niepuste

Złożoność czasowa: $O(\log n)$

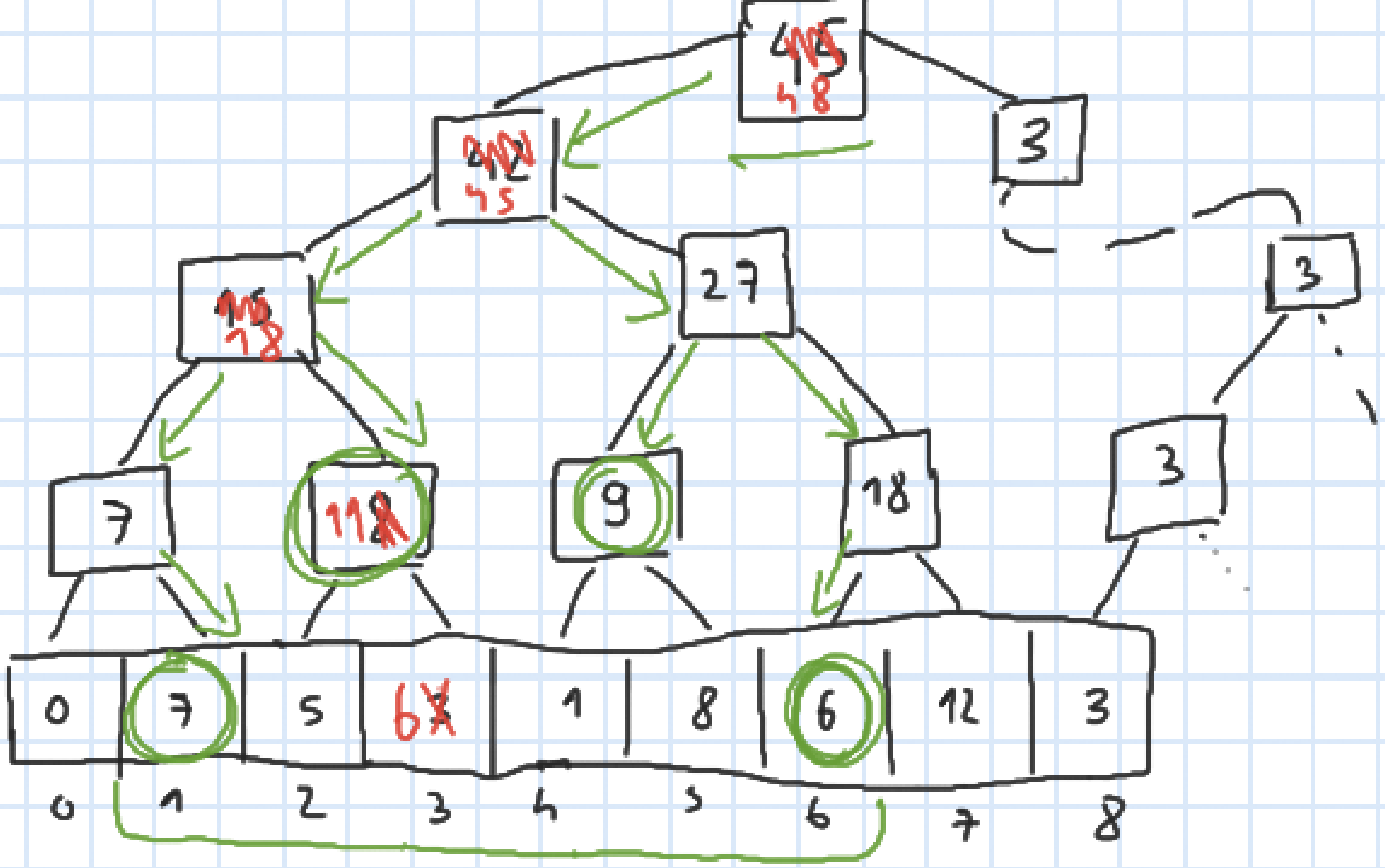
Obliczanie przedziałów, do których należy dany punkt

- znajdujemy przedział bazowy, do którego należy dany punkt
- wędrujemy w górę gromadząc napotkane przedziały

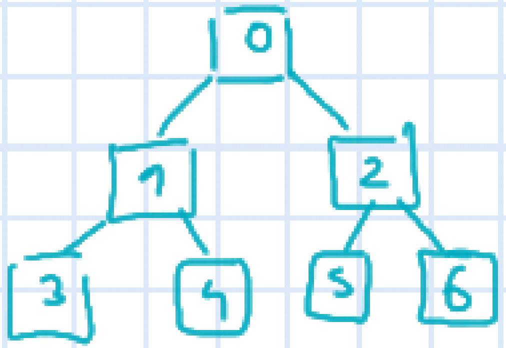
Przykład konstrukcji drzewa przedziałowego, specjalizowanego do innego zadania

Interesuje nas struktura danych przechowująca n liczb indeksowanych od 0 do $n-1$ i oferująca następujące operacje:

- update(i, x) – zamień i-tą liczbę na x
- sum(i, j) – oblicz sumę liczb od i-ej do j-ej



sum$(1,6)$ $=$ $7 + 11 + 9 + 6$



$\text{parent}(x) = \left\lfloor \frac{x-1}{2} \right\rfloor$

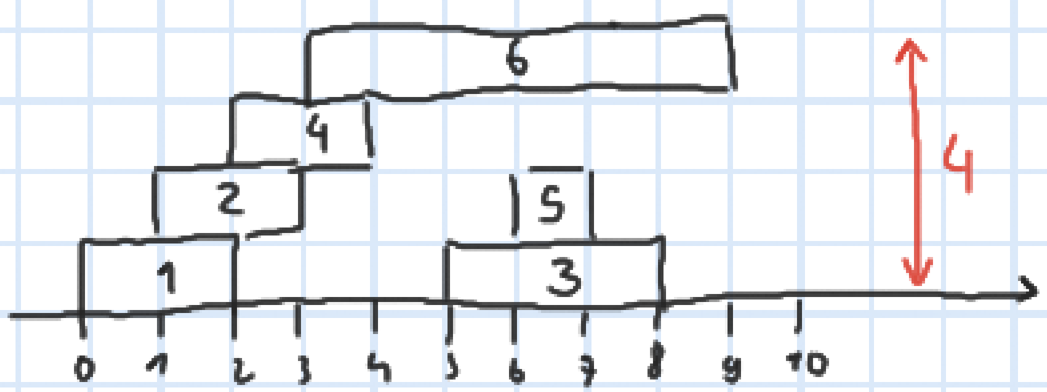
$\text{left}(x) = 2x+1$

$\text{right}(x) = 2x+2$

Update – zmieniamy element tablicy i uaktualniamy sumy w górę drzewa $O(\log n)$

sum(i,j) – analogicznie do ustawiania przedziałów poprzednio

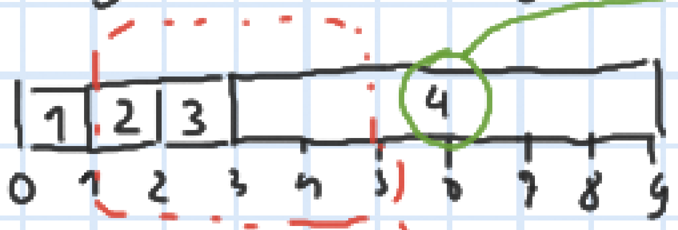
Przykład: Spadające klocki



Zadane klocki spadają w kolejności z góry i dotykają się do najwyższego klocka, który dotykają.

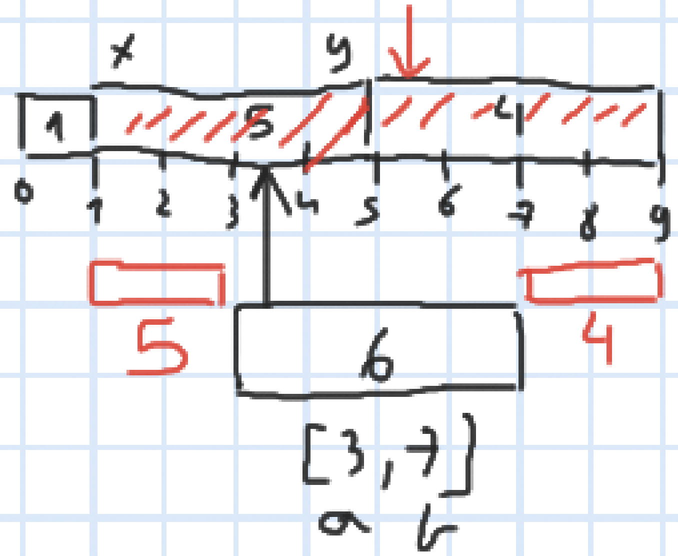
Należy wyznaczyć wysokość powstałej konstrukcji

Stan konstrukcji po pierwszych i klockach można zapisać jako ciąg niezachodzących na siebie przedziałów



wysokość zadanego obszaru

Gdyby spadł klocek [1, 5] to nam ciąg zmienia się na:



Złożoność $O(n \log n)$

Ciąg rozłącznych przedziałów reprezentujących stan konstrukcji przechowujemy w drzewie czerwono-czarnym (sortujemy przedziały względem punktu początkowego)

Jak zrzucamy klocek [a, b]?

① znajdujemy przedział [x, y] zawierający punkt $a + 1/2$ o wys. h

② - usuwamy [x, y] z drzewa
- ustawiamy [a, x] o wys. h

③ - Jeśli [b, y] jest pusty, to powtórz powyższe dla punktu $y + \frac{1}{2}$
- Jak nie jest pusty, ustaw [b, y] z wys. h

④ Ustaw [x, y] z wysokością równą max. wysokości odwiedzonych obszarów + 1

Realizacja w oparciu o drzewa przedziałowe

Dla n klocków mamy najwyżej 2n różnych punktów początku/końca → przenumerowujemy klocki tak, że każdy ma początek i koniec w zakresie od 0 do 2n - 1