

Algorytm i Struktury Danych

Wykład 15

Drewno przedziałowe — interesuje nas struktura przechowywania przedziałów i pozwalająca odnieść listę przedziałów zawierających dany punkt

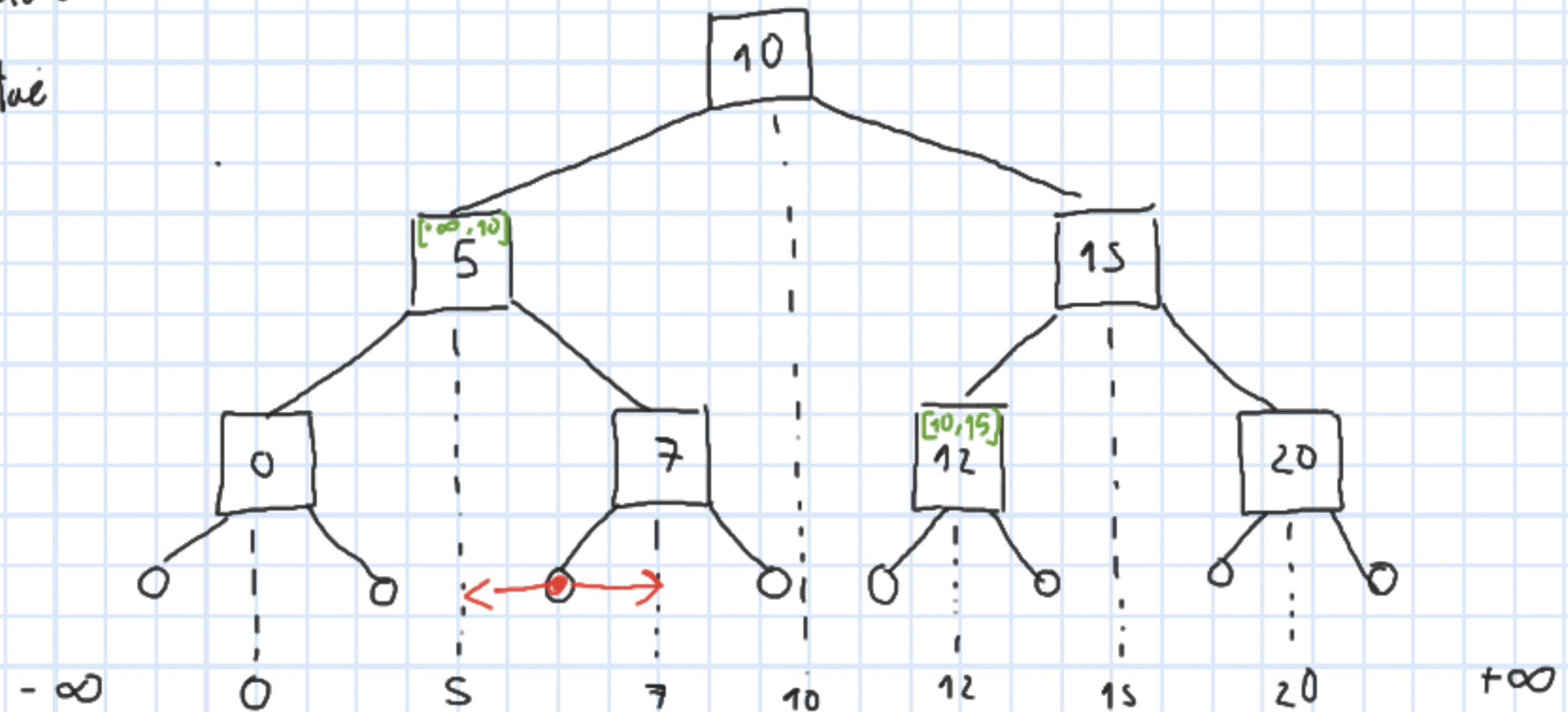
Przykład

Rozważmy przedziały

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



Konstruujemy drewno BST zawierające punkty graniczne przedziałów bazowych

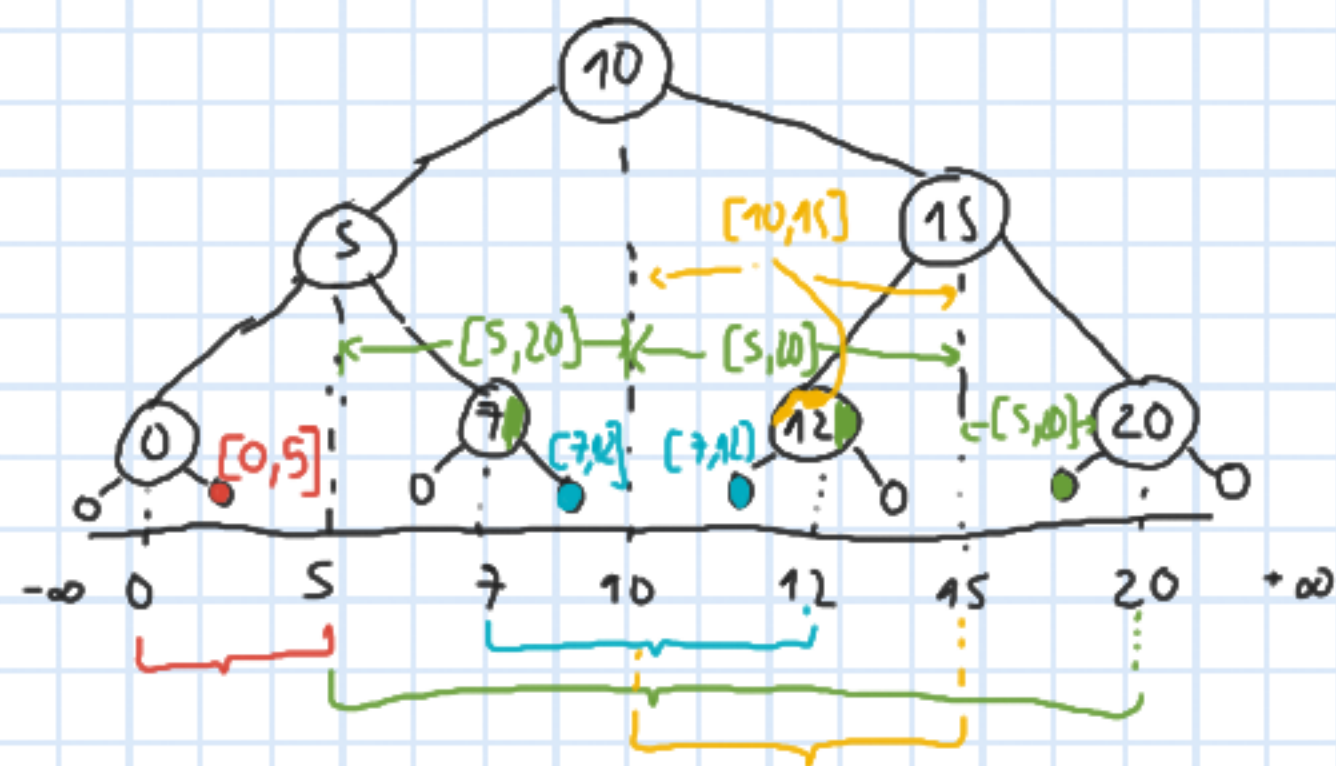


Każdy liść odpowiada za dany przedział bazowy

Każdy węzeł odpowiada za sumę przedziałów bazowych w jego poddrzewie

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

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



Złożoność pamięciowa - każdy przedział na danym

poziomie drzewo występuje najwyżej 2 razy \Rightarrow 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:

- przeszukujemy przedział ze span'em lewego dziecka ustawiamy po lewej stronie (rek.)

- przeszukujemy przedział ze span'em prawego dziecka ustawiamy po prawej

Złożoność czasu: $O(\log n)$

ustawiam u danym
poddrzewie o ile
przeszukałem niepełnie

Obliczanie przedziałów, do których należy dany punkt

- znajdujemy przedział bazy, do którego należy dany punkt

- wędrujemy u góry gromadząc wszystkie przedziały

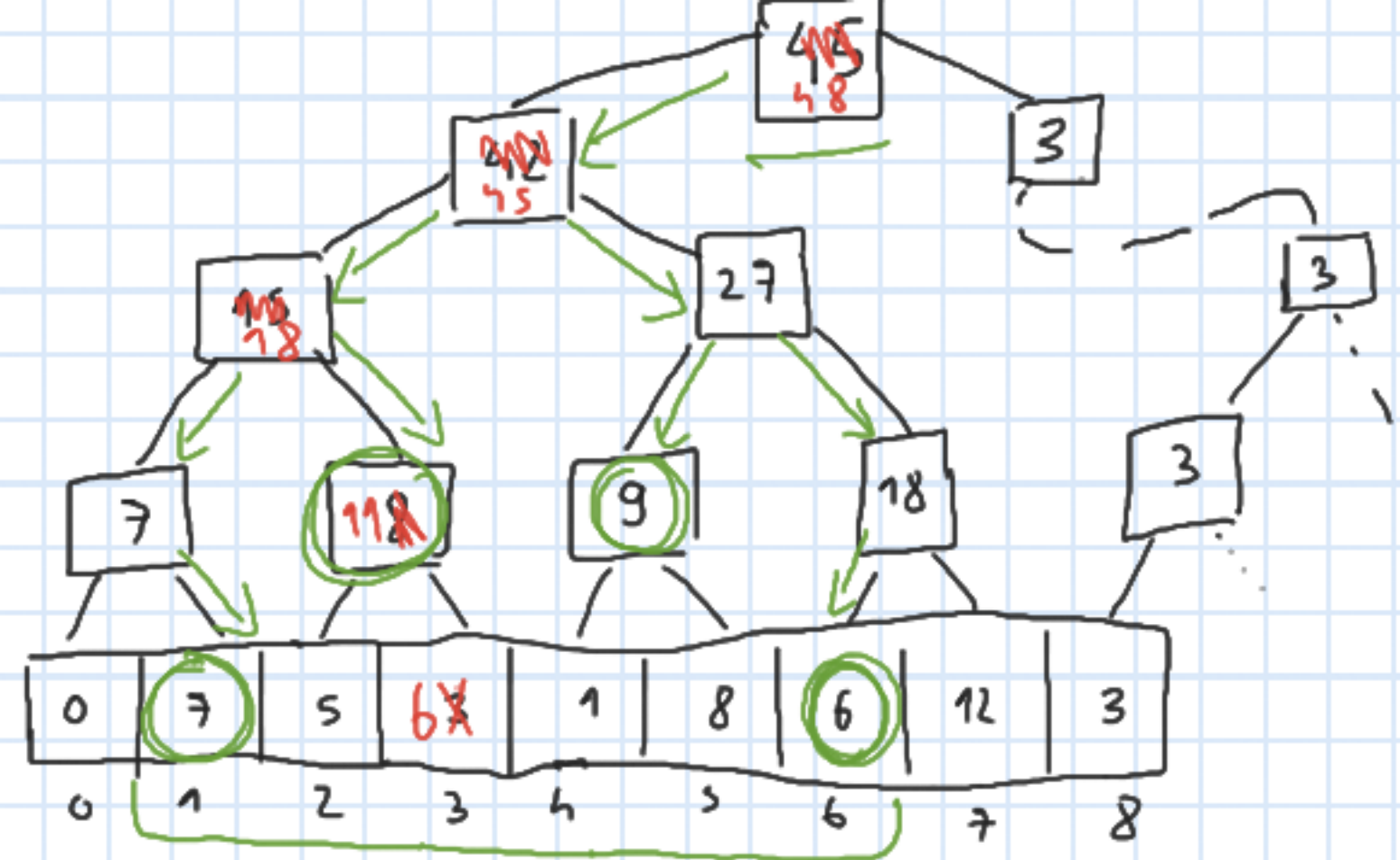
Przykład konstrukcji drzewa przedziałowego, specjalizowanego do innego zadania

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

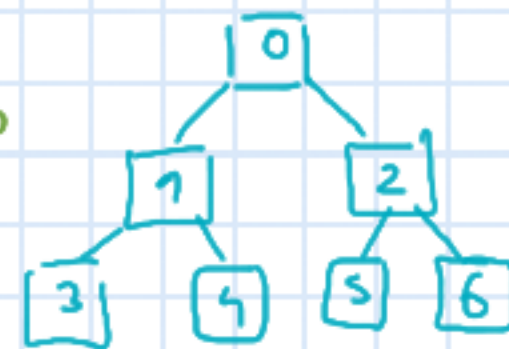
- $update(i, x)$ - zamień i-tą liczbę na x
- $sum(i, j)$ - oblicz sumę liczb od i-ego do j-ego

Update - zmieniamy element tablicy i uaktualniamy sumy u góry drzewa $O(\log n)$

$sum(i, j)$ - analogicznie do ustalania przedziałów poprzednio

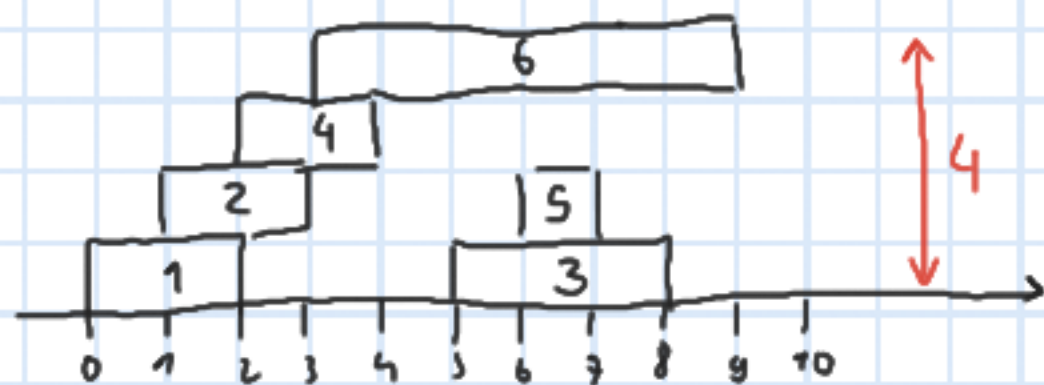


$sum(1,6)$
||
 $7 + 11 + 9 + 6$



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

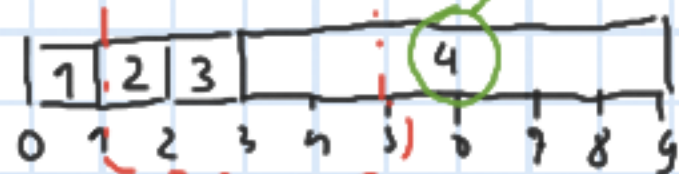
Przykład: Spadające klocki



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

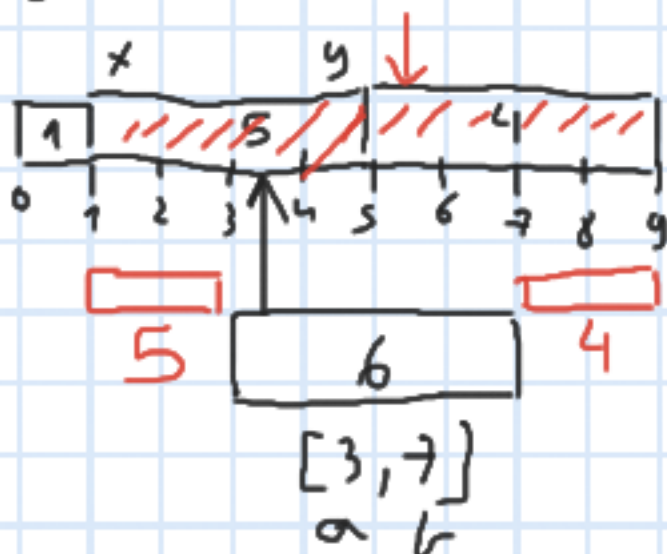
Należy wyznaczyć wysokość
powstałej konstrukcji

Stan konstrukcji po pierwszych i -krokach można zapisać
jako ciąg niezależnych na sobie przedziałów



wysokość zadanego obszaru

Gdyby spadł klocek $[1, 5]$ to nasz ciąg zmienił się na:



Złożoność $O(n \log n)$

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

Jak znaleźć 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 powtórzyć
proces dla punktu $y + 1/2$
- Jeśli nie jest pusty, ustaw $[b, y]$ z wys. h
- ④ Ustaw $[x, y]$ z wysokością równą max.
wysokości odciętych obszarów + 1

Realizacja u oparciu o drzewo przedziałowe

Dla n bloków mamy najwyżej $2n$ różnych punktów początku/konca \rightarrow numerujemy klocki tak, że każdy ma początek i koniec u zakresie od 0 do $2n-1$

