

## Algorytmy i struktury danych (Lista 6)

**Zadanie 1** Jakie informacje przechowujemy w węźle B-drzewa? Podaj definicję B-drzewa.

```
struct BTreeNode:
    int t;           // minimum degree
    int n;           // current number of keys
    int* keys;       // array of keys in non-decreasing order
    bool leaf;       // is it a leaf?
    BTreeNode** children;
end
```

Inne założenia (oprócz zawartych w komentarzach):

1. Węzeł wewnętrzny zawiera  $n + 1$  wskaźników do synów.
2. Klucze rozdzielają dzieci na przedziały ( $n+1$ ).
3. Każdy węzeł różny od korzenia musi mieć co najmniej  $t - 1$  kluczy i co najwyżej  $2t - 1$  kluczy. (korzeń może mieć od 1 do  $2t - 1$  kluczy)
4. Wszystkie liście leżą na tej samej wysokości równym  $h$ .

**Zadanie 2** (2 pkt.) Udowodnij, że żadna z poniższych operacji wykonana na drzewie spełniającym wszystkie warunki B-drzewa, nie prowadzi do ich naruszenia.

- (a) `split_child`, przesuwająca środkowy klucz (medianę) z węzła o  $2t - 1$  kluczach do rodzica, który ma mniej niż  $2t - 1$  kluczy, a klucze i dzieci na prawo od mediany – do nowego brata dodanego po prawej stronie dzielonego węzła. Skoro mogliśmy przejść do dziecka to znaczy, że rodzic był niepełen i może przyjąć przynajmniej jeden klucz. Więc nie narusza się maksymalnej liczby kluczy w węźle. Węzeł dzielony miał  $2t - 1$  kluczy, więc bez tego odjętego zostają nam dwa zestawy po minimalnej liczbie kluczy.
- (b) `unsplit_child` odwrotna do `split_child`, sklejająca dwa sąsiednie węzły o minimalnej liczbie kluczy  $t - 1$  oraz klucz stojący w rodzicu między nimi w jeden nowy węzeł. Zakładamy, że rodzic ma co najmniej  $t$  kluczy lub jest korzeniem.
- (c) `borrow_from_sibling`, rotacja przenosząca do węzła o minimalnej  $t - 1$  liczbie kluczy, który ma prawego brata z co najmniej  $t$  kluczami, klucz stojący w rodzicu między braćmi i wpisująca na jego miejsce jego miejsce pierwszy klucz brata. Jakie operacje na dzieciach należy dodatkowo wykonać?

**Zadanie 3** W B-drzewie o  $t = 10$  podaj wzory i wyniki numeryczne określające:

- (a) ile kluczy może zawierać korzeń (podaj przedział),  
Korzeń zawiera od 1 do 19 kluczy. (max  $2t - 1$ )

- (b) ile dzieci może mieć korzeń (podaj przedział),  
Korzeń może mieć od 2 do 20 dzieci. (min  $t$  max  $2t$ )
- (c) ile kluczy może mieć potomek korzenia (podaj przedział),  
Potomek korzenia może mieć od 9 do 19 kluczy. (min  $t - 1$  max  $2t - 1$ )
- (d) ile dzieci może mieć potomek korzenia (podaj przedział),  
Potomek korzenia może mieć od 10 do 20 dzieci. (min  $t$  max  $2t$ )
- (e) ile maksymalnie węzłów może być na  $k$ -tym poziomie (przyjmując, że korzeń to poziom 0)  
Na  $k$ -tym poziomie może być maksymalnie  $(2t)^k$  węzłów.
- (f) ile łącznie kluczy może być na  $k$ -tym poziomie (podaj przedział).  
Nie licząc korzenia dla którego minimum to 1 klucz to na  $k$ -tym poziomie może być od  $2(t - 1)t^{k-1}$  do  $(2t - 1)(2t)^k$  kluczy. (min  $(2min)t^{k-1}$  max  $(max)t^k$ )

**Zadanie 4** Jaka jest minimalna, a jaka maksymalna liczba kluczy w B-drzewie mającym  $h$  poziomów, przy ustalonej wartości parametru  $t$  (patrz Cormen).

Minimalna:

Gdy korzeń zawiera 1 klucz, a na każdym z pozostałych poziomów każdy węzeł zawiera  $t - 1$  kluczy (minimum).

Na poziomie 1 mamy 2 węzły  $\rightarrow 2(t - 1)$  kluczy.

Z czego wynika, że na kolejnym poziomie mamy  $2t$  węzłów, na następnym  $2t^2$  itd.

W takim układzie na  $h$  poziomie mamy  $2t^{h-1}$  węzłów, a każdy z nich zawiera  $t - 1$  kluczy.

Stąd:

$$\begin{aligned}
 n &\geq 1 + 2(t - 1)(2t + 2t^2 + \dots + 2t^{h-1}) \\
 &= 1 + (t - 1) \sum_{i=1}^h 2t^{i-1} \\
 &= 1 + 2(t - 1) \frac{t^h - 1}{t - 1} \\
 &= 2t^h - 1
 \end{aligned}$$

Maksymalna:

Gdy korzeń ma maksymalną ilość kluczy  $2t - 1$ , a na każdym z pozostałych poziomów każdy węzeł zawiera  $2t - 1$  kluczy (maksimum).

Na poziomie 1 mamy max  $2t - 1$  kluczy razy węzły  $2t \rightarrow (2t - 1)2t$ .

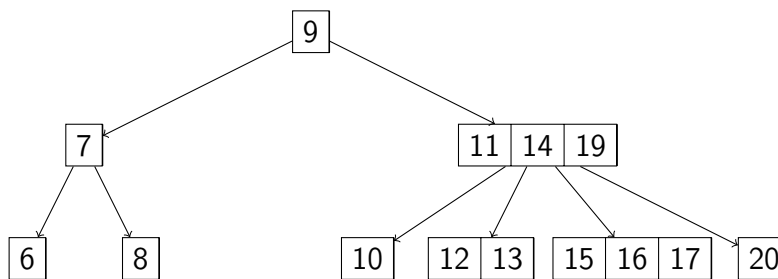
Z czego wynika, że na kolejnym poziomie mamy  $(2t - 1)2t^2$  kluczy, na następnym  $(2t - 1)2t^3$  itd.

W takim układzie na  $h$  poziomie mamy  $(2t - 1)2t^h$  kluczy, a każdy z nich zawiera  $2t - 1$  kluczy.

Stąd:

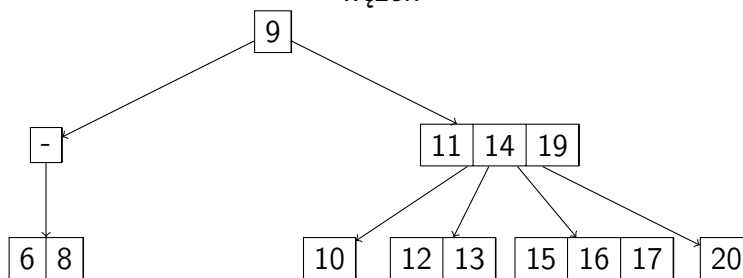
$$\begin{aligned}
n &\leq (2t - 1) + (2t - 1)(2t + (2t)^2 + \dots + (2t)^h) \\
&= (2t - 1) + (2t - 1) \sum_{i=1}^h (2t)^i \\
&= (2t - 1) \sum_{i=0}^h (2t)^i \\
&= (2t - 1) \frac{(2t)^{h+1} - 1}{2t - 1} \\
&= (2t)^{h+1} - 1
\end{aligned}$$

**Zadanie 5** Podano na rysunku B-drzewo o  $t = 2$ :

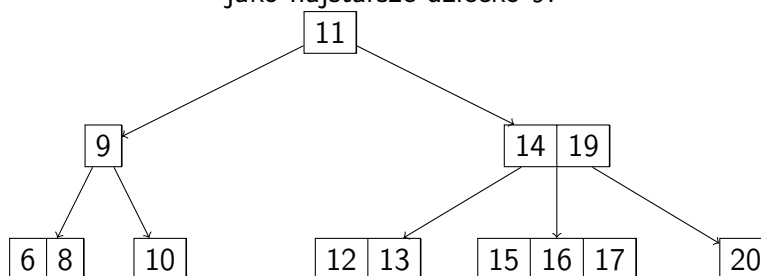


- usuń z tego drzewa 7.
- do drzewa widocznego powyżej dodaj 18.

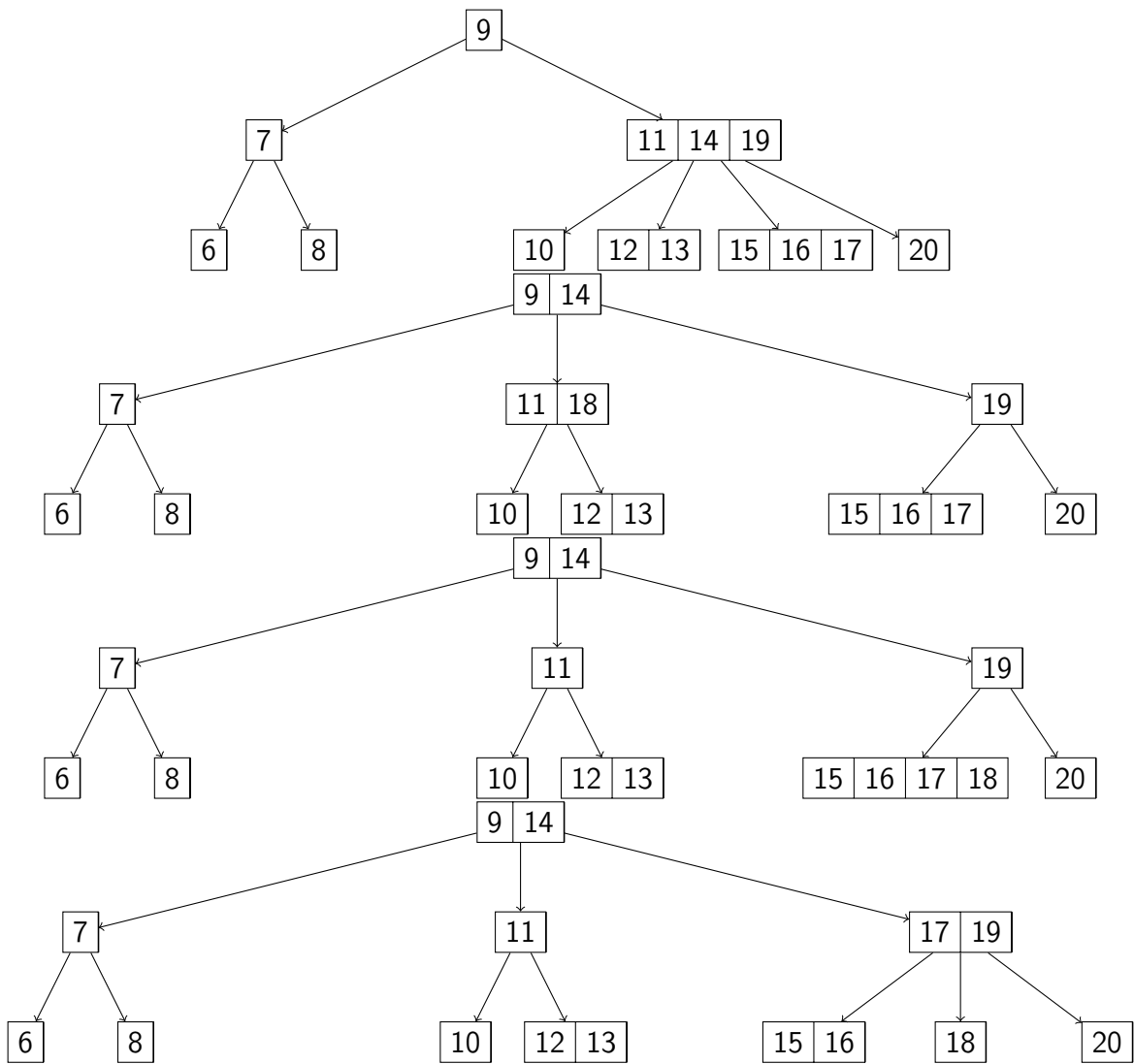
Próbujemy zastąpić 7 przez klucz z dziecka, ale brakuje nam liścia, więc łączymy dzieci (6 i 8) w jeden węzeł.



Nowym rodzicem będzie 9 a korzeniem 11 pożyczona z brata 7. Najmłodsze dziecko 11 przierzucamy jako najstarsze dziecko 9.

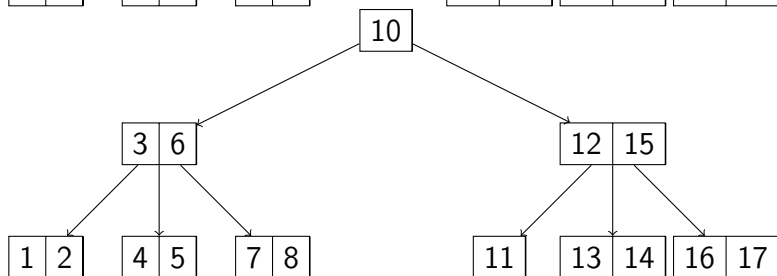
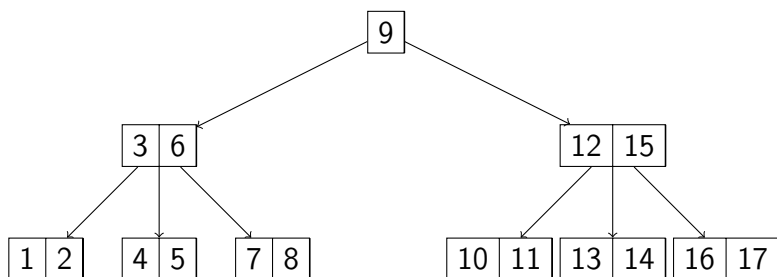


Dodanie 18:



**Zadanie 6** (2 pkt.) Do pustego B-drzewa o  $t = 2$  wstaw kolejno 22 litery swojego imienia i nazwiska oraz adresu. Następnie usuń w tej samej kolejności w jakiej były wstawiane.

**Zadanie 7** Narysuj B-drzewo o  $t = 3$  zawierające dokładnie 17 kluczy na trzech poziomach: korzeń, jego dzieci i wnuki. Następnie usuń z tego drzewa korzeń.



11 nie spełnia sama warunku B-drzewa, więc musimy pożyczyć od rodzica 12, jednak wtedy z 15 dzieje się to samo. Nie mamy jak pożyczyć od rodzica ani od brata. Dlatego musimy złączyć je w jeden węzeł.

