

# Struktury Danych i Złożoność Obliczeniowa

## ĆWICZENIA 2

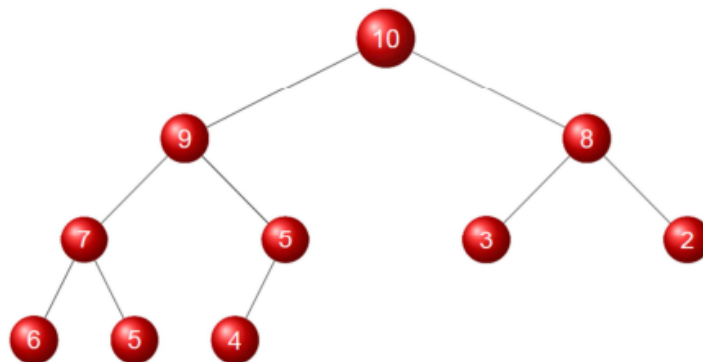
### Kopiec:

Kopiec to drzewo binarne, czyli takie, gdzie każdy rodzic ma maksymalnie 2 potomków. W kopcu wszystkie poziomy głębokości poza ostatnim są w pełni wypełnione. Ostatni poziom głębokości zawsze jest dosunięty do lewej strony.

Istnieją dwa rodzaje kopców:

- *Maksymalny* - gdzie wartość rodzica jest większa lub równa wartości potomków
- *Minimalny* - gdzie wartość rodzica jest mniejsza lub równa wartości potomków

Element w pierwszym rzędzie nazywamy korzeniem kopca.



### Złożoności dla kopca:

- *Dodawanie elementu* -  $O(\log n)$
- *Usuwanie elementu* -  $O(\log n)$
- *Tworzenie kopca* -  $O(n \log n)$  lub  $O(n)$  dla algorytmu Floyda

- *Wyszukiwanie* -  $O(n \log n)$ , bo musimy zdejmować z kopca elementy, aż do znalezienia

#### **Dodawanie do kopca:**

Dodawanie do kopca polega na włożeniu elementu na pierwszą wolną pozycję w ostatnim rzędzie. Następnie przeprowadzamy naprawę kopca w górę, aby przywrócić odpowiednie zależności między rodzicami a potomkami. W związku z wymogiem naprawy kopca operacja dodania elementu ma złożoność:  $O(\log n)$ .

#### **Usuwanie z kopca:**

W miejsce usuniętego elementu wstawiamy ostatni dodany do kopca element. Następnie naprawiamy kopiec w dół.

#### **Implementacja kopca w tablicy:**

Najpopularniejszy sposób komputerowej implementacji kopca. Taka reprezentacja ma pewne ciekawe zależności między indeksami swoich elementów:

*INDEKS RODZICA*

$$r = \lfloor \frac{1}{2}(i - 1) \rfloor$$

*INDEKSY POTOMKÓW*

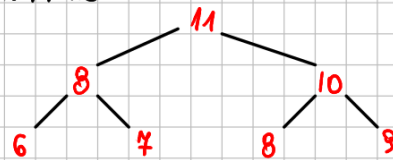
$$p_1 = 2i + 1 \text{ lub } p_2 = 2i + 2$$

Algorytm dodawania elementu:

- 1) Określ rodzica, jeżeli nie ma rodzica to zakończ
- 2) Sprawdź, czy rodzic i potomek spełniają relację
  - Jeżeli tak, to zakończ
  - Jeżeli nie, to zamień miejscami rodzica z potomkiem
- 3) Powtórz

0	1	2	3	4	5	6
11	8	9	6	7	8	10
11	8	10	6	7	8	9
11	8	10	6	7	8	9

WYNIK:



## Algorytm usuwania elementu:

- 1) Usun element  $i$  w jego miejsce wpisz element z ostatniej pozycji.
- 2) Oznacz rodzica i jego potomków.
- 3) Sprawdź czy zachowane są relacje rodzic-potomek:
  - Jeżeli nie, to zamień miejscami rodzica z mniejszym (minimalnym) lub większym (maksymalnym) potomkiem
  - W przeciwnym wypadku przejdź do kolejnego elementu
- 4) Przejdź do 2

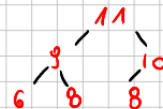
$$\frac{4-1}{2} = 1,5$$

naprawa w dół

0	1	2	3	4	5	6
11	8	10	6	X	8	3
11	8	10	6	9	8	
11	8	10	6	9	8	
11	8	10	6	9	8	
11	8	10	6	9	8	
11	8	10	6	9	8	

naprawa w górę

0	1	2	3	4	5	6
11	8	10	6	X	8	3
11	8	10	6	9	8	
11	8	10	6	9	8	
11	8	10	6	9	8	
11	8	10	6	9	8	
11	8	10	6	9	8	



## Algorytm naprawy kopca Floyda:

- 1) Bieremy ostatniego rodzica
- 2) Sprawdzamy, czy zachowane są relacje rodzic-potomkowie
  - Jeżeli nie, zamieniamy miejscami rodzica z mniejszym (minimalny) lub większym (maksymalny) potomkiem. Zachowujemy zaznaczenie ze przemieszczonym rodzicu
  - Jeżeli tak, to idziemy do poprzedniego rodzica.
- 3) Powrót do kroku 2

0	1	2	3	4	5	6
11	8	10	6	X	9	9
11	8	10	6	9	8	
11	8	10	6	9	8	
11	9	10	6	8	8	
11	9	10	6	8	8	

Ostatni rodzic:

$$\left\lfloor \frac{5-1}{2} \right\rfloor = \left\lfloor \frac{4}{2} \right\rfloor = 2$$

