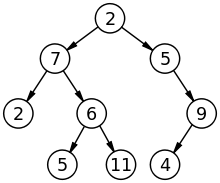
**Kolejka priorytetowa implementacja za pomocą kopca binarnego**

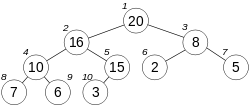
**Czym jest kolejka priorytetowa – jest to kolejka, którą elementy opuszczają nie według kolejności dodania (FIFO) a według pewnego priorytetu. Jednym ze sposobów implementacji kolejki priorytetowej jest kopiec binarny, którego właściwości omówimy poniżej odpowiadając na cztery pytania:**

1. Co to jest kopiec binarny – jest to kompletne drzewo binarne spełniające warunek kopca.
2. Co to jest drzewo binarne – jest to drzewo, w którym każdy węzeł ma co najwyżej dwóch potomków.
3. Co to znaczy, że drzewo jest kompletne – oznacza to, że w drzewo posiada zapełnione wszystkie poziomy nie licząc ostatniego, a w ostatni poziom jest wypełniony od lewej do prawej (bez luk).
4. **Warunek kopca: dzieci większe (lub mniejsze) od rodzica. To czy większe czy mniejsze zależy od implementacji.**

Przykłady (z Wikipedii):



To jest drzewo. Ma ono 4 poziomy (numeracja od 0 do 3). Korzeniem jest 2. Nie jest ono kompletne (np. na poziomie nr 2 mamy liść – węzeł z 2), 5 na poziomie 1 ma tylko jednego potomka itd.



To jest drzewo binarne kompletne (w tej wersji w korzeniu jest element największy).

Można go przechowywać w macierzy (numery obok węzłów to indeksy elementów w macierzy – z tym że indeksowanie na rysunku przebiega od 1 a nie od 0 jak w c++).

**Niezbędne operacje:**

1. Musimy mieć możliwość dodania elementu do drzewa.
2. Musimy mieć możliwość usunięcia elementu w korzeniu
3. Po dwóch powyższych operacjach musimy mieć możliwość naprawy struktury drzewa.

**Kolejka priorytetowa a kopiec:**

1. Usunięcie elementu z kolejki to usunięcie korzenia (i naprawa stanu drzewa)
2. Dodanie elementu do kolejki to dodanie elementu do kopca (czyli dodanie elementu na pierwsze wolne miejsce na ostatnim poziomie oraz poprawa warunku drzewa).

**Szybkość operacji:**

Dodanie elementu O(log n)

Sprawdzenie najmniejszego (lub największego – w zależności od implementacji) O (1)

Usunięcie najmniejszego (największego) elementu z kolejki O (log n) – bo po usunięciu trzeba naprawić drzewo.

Aby znaleźć więcej informacji o kopcach binarnych oraz jak je implementować można rzucić okiem np. tutaj:

http://math.uni.wroc.pl/~jagiella/p2python/skrypt\_html/wyklad9.html#kopce\_kopcowanie