

# Laboratorium AiSD

## Lista 7

### Wyszukiwanie, kolejki priorytetowe, tablice mieszające

Proszę pamiętać, że **część rozwiązania** zadania stanowi również **zestaw testów** zaimplementowanych algorytmów i/lub struktur danych. Dodatkowo, proszę zwracać uwagę na **powtarzające się fragmenty** kodu i wydzielać je do osobnych funkcji/klas.

1. Zdefiniuj klasę implementującą strukturę **binarnego kopca maksymalnego** (ang. *binary max-heap*) ***BinaryHeap*<T>** zaimplementowanego na drzewie.

**Interfejs kopca** powinien udostępniać trzy operacje:

- *void clear()* – czyszczącą kopiec (usuającą wszystkie elementy),
- *void add(T element)* – wstawiającą nowy element do kopca,
- *T maximum()* – zwracającą maksymalny element kopca wraz z jego usunięciem.

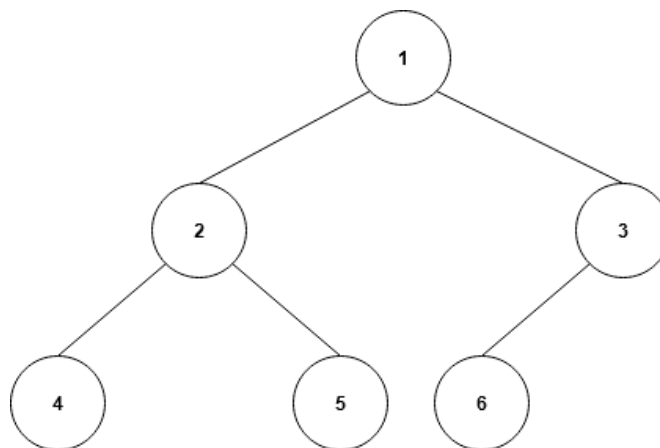
Opis budowy kopca opisano na **wykładzie 6**.

W implementacji węzły **nie mogą** mieć referencji na rodzica!

Założ, że **null** nie jest poprawną wartością.

#### Wskazówka:

Główną trudność w zadaniu stanowi sposób odnalezienia właściwej pozycji węzła w drzewie. Numerując jednak węzły w sposób jak na Rysunku 1, można zauważyć, że istnieje zależność między ścieżką, a indeksem. Przykładowo,  $6_{(10)} = 110_{(2)}$ .



Rysunek 1 Przykładowy kopiec binarny (w węzłach podano ich indeksy!)