

# Drzewa

Program zaliczeniowy nr 1 z przedmiotu *Algorytmy i Struktury Danych*  
semestr letni 2018/2019

## Wstęp

Historię ewolucji organizmów zazwyczaj reprezentuje się za pomocą *drzew filogenetycznych*, w których liście reprezentują gatunki współczesne, zaś wierzchołki wewnętrzne wspólnych przodków liści znajdujących się w zawieszonych w nich poddrzewach. Ponadto krawędziom takiego drzewa można przypisać wagi reprezentujące *odległości ewolucyjne* między gatunkami z końców krawędzi. Odległość między dwoma dowolnymi wierzchołkami jest wówczas sumą wag krawędzi ze ścieżki łączącej te wierzchołki. Niekiedy wymaga się, żeby gatunki współczesne (czyli liście) były jednakowo odległe od swego wspólnego przodka (czyli korzenia).

## Zadanie

Należy napisać moduł *ImieNazwisko.py* (najlepiej własne;) zawierający klasy `PhylTree` oraz `PhylNode` implementujące odpowiednio drzewa filogenetyczne oraz ich wierzchołki. Każdy wierzchołek przechowuje wagę wchodzącej krawędzi, czyli odległość od swojego rodzica. Wierzchołki mogą mieć dowolną liczbę dzieci.

Klasa `PhylNode` powinna zawierać następujące metody:

- `__init__(distance=None, children=[])` utwórz wierzchołek z dziećmi z listy `children`, odległy o `distance` od swojego rodzica w drzewie
- `get_children()` – zwróć listę dzieci
- `get_distance()` – zwróć odległość wierzchołka od rodzica (lub `None`, jeśli ta nie została ustalona)
- `set_distance(distance)` – zmień odległość wierzchołka od jego rodzica na `distance`

Klasa `PhylTree` powinna zawierać następujące metody:

- `__init__(node)` – utwórz drzewo z wierzchołkiem `node` w korzeniu
- `root()` – zwróć korzeń drzewa
- `max_leaf_distance()` – zwróć maximum z łącznych odległości od korzenia do liści
- `min_leaf_distance()` – zwróć minimum z łącznych odległości od korzenia do liści
- `average_leaf_distance()` – zwróć średnią z łącznych odległości od korzenia do liści
- `elongate_distances()` – zmodyfikuj wagi krawędzi tak, aby wszystkie łączne odległości od korzenia do liści były równe, przy czym żadna z wag nie może zostać zmniejszona
- `truncate_distances()` – zmodyfikuj wagi krawędzi tak, aby wszystkie łączne odległości od korzenia do liści były równe, przy czym żadna z wag nie może zostać zwiększona
- `unify_distances()` – zmodyfikuj wagi krawędzi tak, aby wszystkie łączne odległości od korzenia do liści były równe

## Ocena

Rozwiązanie zadania powinno zawierać kod programu z komentarzami. Implementacja powinna działać sprawnie (na przeciętnym komputerze PC czas wykonania metody na drzewie o kilkunastu liściach nie może przekraczać kilku sekund). Rozwiązanie jest *optymalne*, jeśli modyfikacje wykonywane w 3 ostatnich metodach są możliwie niewielkie, tzn. jeśli suma wartości bezwzględnych zmian wag jest zminimalizowana.

Ocena rozwiązania:

**18 pkt.** pełne i optymalne rozwiązanie

**12 pkt.** pełne rozwiązanie

**7 pkt.** rozwiązanie bez metod `elongate_distances()`, `truncate_distances()` i `unify_distances()`