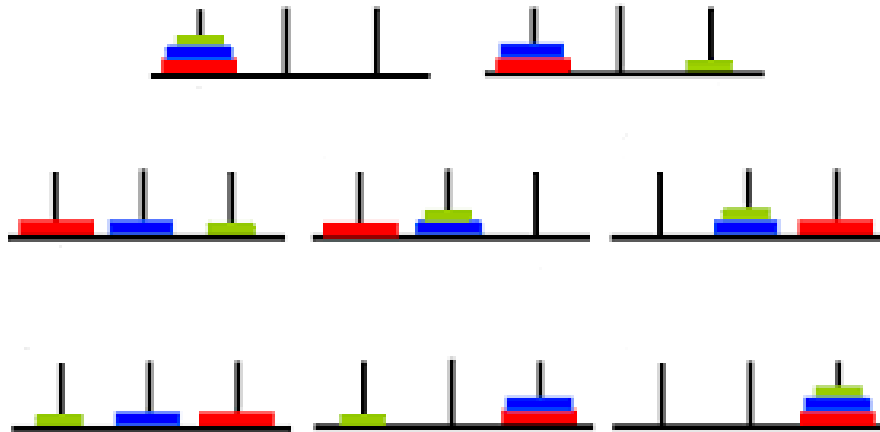


Dany jest problem wież Hanoi polegający na odbudowaniu, z zachowaniem kształtu, wieży z krążków o różnych średnicach, przy czym podczas przekładania dozwolone jest posługiwanie buforem w postaci dodatkowego słupka. Operacje odbywają się przy założeniu, że nie wolno położyć krążka o większej średnicy na mniejszy, ani przekładać kilku krążków jednocześnie.



Rysunek 1: Wizualizacja algorytmu dla trzech krążków.

Zadanie 1 (0.75 pkt) Zaimplementuj rekurencyjny algorytm rozwiązujący problem wież Hanoi. Algorytm powinien zliczać ilość potrzebnych kroków oraz wypisywać kolejno ruchy prowadzące do rozwiązania problemu.

Poniżej przedstawiono przykładowy zapis algorytmu w pseudokodzie, gdzie zmienna n to liczba krążków, zmienne $sour$, $dest$, $buff$ to odpowiednio słupki początkowy, docelowy, dodatkowy.

```
Hanoi(n, sour, dest, buff):
  IF n==1 :
    Move disk from sour to dest
  Hanoi (n-1, sour, buff, dest)
  Move disk from sour to dest
  Hanoi (n-1, buff, dest, sour)
```

Zadanie 2 (0.75 pkt) Zaimplementuj iteracyjny algorytm rozwiązujący problem wież Hanoi. Algorytm powinien zliczać ilość potrzebnych kroków oraz wypisywać kolejno ruchy prowadzące do rozwiązania problemu.

Poniżej przedstawiono algorytm zapisany w pseudokodzie, gdzie zmienna n to liczba krążków, i numer kolejnego kroku, zmienne $sour$, $dest$, $buff$ to odpowiednio słupki początkowy, docelowy, dodatkowy.

```
Hanoi(n, sour, dest, buff):  
    WHILE (sour != Null OR buff !=Null):  
        IF i%3 == 1:  
            Possible move disk between sour and dest  
        IF i%3 == 2:  
            Possible move disk between sour and buff  
        IF i%3 == 0:  
            Possible move disk between buff and dest
```

Uwaga: Należy sprawdzić, w którą stronę ruch jest możliwy i wskazać z którego słupka na który powinien zostać wykonany ruch.

Zadanie 3 (0.5 pkt) Sprawdź poprawność obu algorytmów porównując kolejno wykonywane ruchy oraz ich ilość. Porównaj szybkość działania obu algorytmów w zależności od liczby krążków. Czy oba algorytmy są optymalne?
