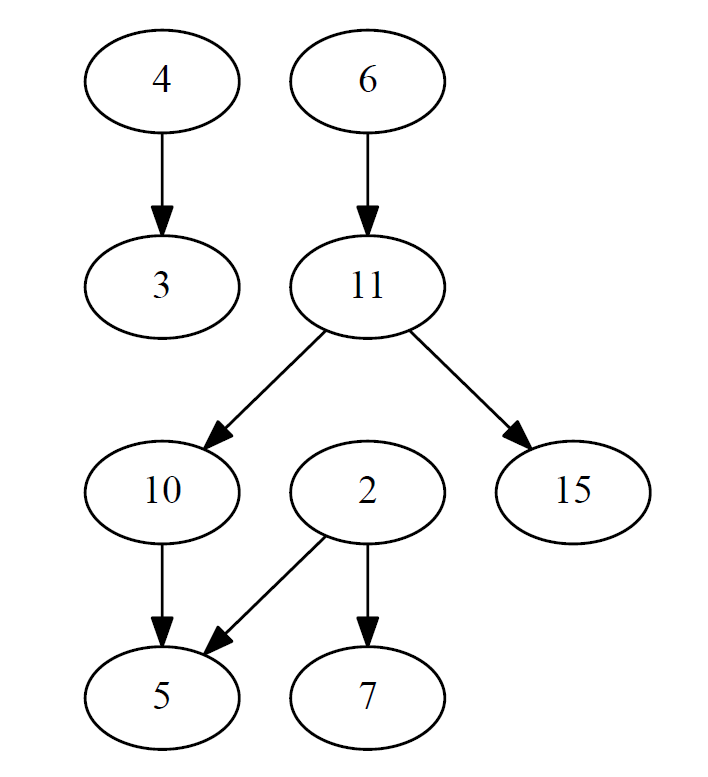
Julia Grzegorzewska, Wiktoria Fimińska

Link do repozytorium: https://github.com/grzesiaaa/Algorytmy\_lista7

**RAPORT LISTA 7**

**Zadania 1, 2**

Do klasy Graph z wykładu dodałyśmy metodę createDot, generującą reprezentację grafu z języku dot korzystając z programu graphviz. Przykładowe wywołanie:



**Zadania 3, 4**

Rozbudowałyśmy klasę Graph o metody przeszukiwania wszerz i w głąb (wykład).

W zad 4 mamy zmodyfikowaną funkcję przeszukiwania w głąb, tak aby sortowała graf topologicznie.

**Zadanie 5**

Korzystając z przeszukwiania wszerz liczymy najkrótsze ścieżki od podanego węzła do pozostałych. Wynik zwracamy w postaci słownika z dystansem do danego węzła.



**Zadanie 6**

Stworzyłyśmy klasę, która rozwiąże problem misjonarzy i kanibali przy podanych ilościach każdej z grup. Pierwsze dwie wartości to ilość kolejno misjonarzy i kanibali na lewym brzegu, środkowa wartość oznacza, po której stronie znajduje się łódź, a dwie ostatnie misjonarzy i kanibali po prawej stronie rzeki. W metodzie *move\_boat* określamy jakie możliwości mogą zajść na łodzi: czy dwóch kanibali lub misjonarzy, czy kanibal i misjonarz, czy jedn misjonarz lub kanibal. Następnie w zależności czy łódka jest po prawej czy po lewej, sprawdza, czy ilość kanibali nie przekracza ilości misjonarzy. W metodzie *creating\_solutions* korzystamy z kolejki, aby stworzyć graf, a przy pomocy *createDot1* ładnie wizualizujemy z pomocą biblioteki graphviz i zapisuje w pliku „solutions.gv”.

**Zadanie 7**

Mamy odmierzyć podaną ilość wody przy pomocy dwóch kanistrów o podanych objętościach (zakładamy że pojemność pierwszego większa od drugiego). Jeśli któryś z kanistrów ma mniejszą pojemność niż docelowa to nie uda się tego zrobić.

W pierwszym kroku napełniamy 1 lub 2 kanister do pełna. Następnie w zależności od tego ile płynu mamy w kanistrach przelewamy go z jednego do drugiego lub wylewamy. Na koniec szukamy najkrótszej ścieżki od stanu początkowego (0,0) do końcowego, czyli gdy osiągniemy sytuację (goal,0). Przykładowe wywołanie dla can1 = 4, can2 = 3, goal = 2:

