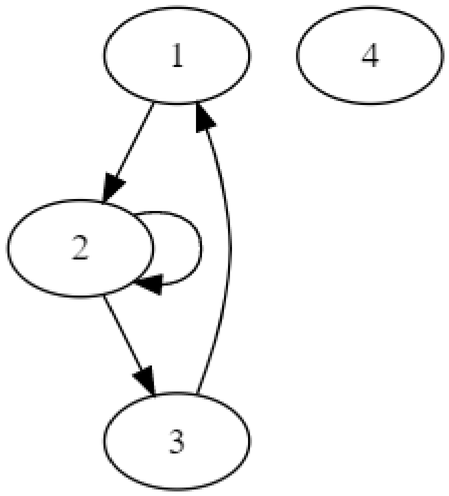
Algorytmy kombinatoryczne w bioinformatyce − Zadanie II

Opis zastosowanego formatu zapisu grafu:
Graf przechowywany jest w pliku tekstowym w postaci zbioru skierowanych krawędzi, każda linia zawiera reprezentację osobnego łuku. Krawędź przedstawiona jest w formacie *x->y,* gdzie *x* i *y* oznaczają numery wierzchołków poprzednika i następnika krawędzi, strzałka symbolizuje zwrot krawędzi. Dopuszczalne jest wystąpienie kilku kopii tych samych krawędzi, zapis taki będzie traktowany jako krawędź wielokrotna. Krawędź w postaci *x->x* oznacza pętlę własną wierzchołka *x*, samo *x* natomiast wierzchołek izolowany.

Przykładowy plik wejściowy i jego reprezentacja graficzna:

1->2
2->3
2->2
3->1
4



Opis algorytmu:

1. **Wczytanie grafu z pliku do pamięci.**
   Skanujemy plik linia po linii, po wczytaniu tworzymy obiekt reprezentujący wierzchołek, zawiera on indeks wierzchołka (etykietę) oraz listę następników i poprzedników.

2. **Sprawdzenie czy graf jest sprzężony zgodnie z poniższą definicją.**
   $$(N^+(x) \bigcap N^+(y) \neq \emptyset) \Rightarrow (N^+(x) = N^+(y))$$
   Sprawdzenie następuje dla par wszystkich wierzchołków tzn. wszystkich dwuelementowych kombinacji bez powtórzeń zbioru wierzchołków.
   Dla każdej wygenerowanej pary wyszukiwany jest element wspólny zbiorów następników, jeżeli takowy istnieje (zachodzi implikacja) wykonywana jest kolejny blok sprawdzający równość zbiorów następników. Jeżeli zbiory są równe przechodzimy do sprawdzania kolejnych wierzchołków, w przeciwnym wypadku przerywamy wykonywanie ponieważ graf nie jest sprzężony.

3. **Jeżeli graf jest sprzężony sprawdzenie czy jest liniowy**
   Sprawdzanie czy graf jest liniowy zachodzi równolegle ze sprawdzaniem sprzężenia. Jeżeli implikacja z powyższej definicja zachodzi następuje sprawdzenie zbiorów poprzedników, jeżeli nie mają one wspólnych elementów oznacza to że graf jest liniowy.

4. **Przekszatłcenie grafu w jego graf oryginalny**
   Jeżeli graf jest grafem sprzężonym, dochodzi do przekształcenia go w jego graf oryginalny (H):
   Do tego celu wykorzystujemy funkcję w której kolejno robimy:
   - Każdy wierzchołek z G zamieniamy najpierw na osobny łuk w H.
   - Dokonujemy przeindeksowania łuków w grafie sprzężony w wierzchołki z grafu oryginalnego (H).
   - Następnie należy odtworzyć jeden do jednego połączenia obecne w grafie sprzężonym poprzez wykonanie przejścia ze zgodnym zwrotem łuku X do łuku odpowiadającego Y. Zatem konieczne jest skompresowanie wierzchołków dla rozłącznych łuków np. (1,2) i (3,4) poprzez zindeksowanie w zbiorze wszystkich wystąpień 3 na 2.
   - Powyższą operacje powtarzamy dla wszystkich łuków z grafu sprzężonego (G).
   - Tak samo należy postąpić dla wszystkich łuków z G po kolei.
   - Wierzchołki zostaną zindeksowane po kolei, ze względu na zastosowany format zapisu plików (Kolumna I jest chronologiczna).

- Algorytm kończymy zapisem grafu wynikowego H do pliku tekstowego dowolnie zlokalizowanego (innego niż wejściowy), w tym samym formacie.

5. **Oszacowanie złożoności algorytmu:**

Algorytm ze względu na swoją specyfikację posiada złożoność równą:
- Dla funkcji sprawdzającej graf sprzężony = O(n^4)
- Dla funkcji sprawdzającej graf liniowy = O(n^4)
- Dla funkcji przekształcającej graf sprzężony w graf oryginalny = O(n^2)

- Złożoność pesymistyczna algorytmu:
  - graf wejściowy jest grafem sprzężonym oraz liniowym
- Złożoność optymistyczna algorytmu:
  - graf wejściowy nie jest ani grafem sprzężonym, ani liniowym

6. **Opis przeprowadzonych testów wraz z wizualizacją przykładowych transformacji przeprowadzonych przez algorytm:**

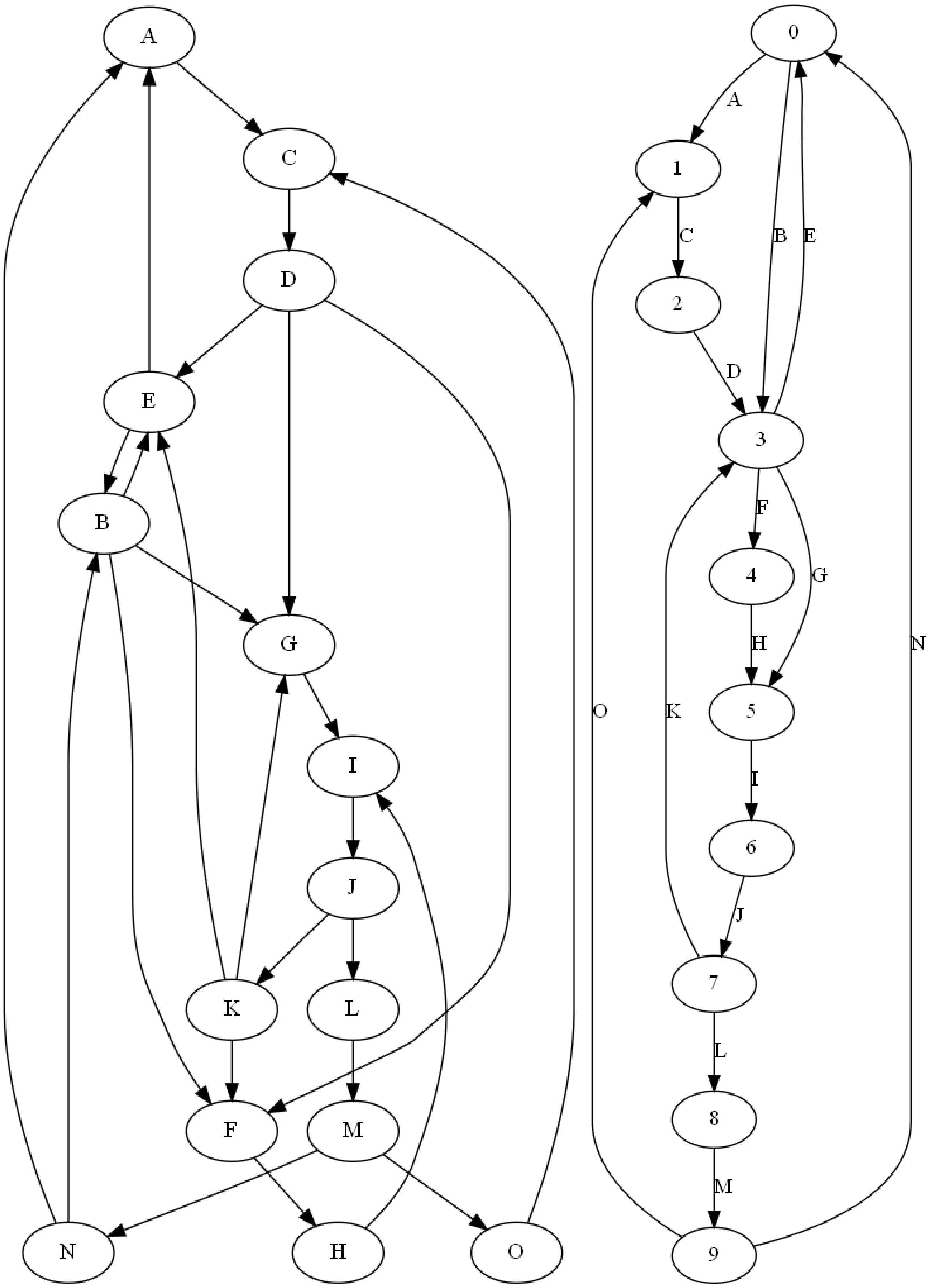
Testy zostały przeprowadzone na grafach, które można transformować na graf oryginalny po uprzednim sprawdzeniu, czy jest on sprzężony oraz liniowy w programie skonstruowanym na postawie algorytmu. Zwizualizowane zostały przykładowe transformacje widoczne poniżej:

(Aby zweryfikować poprawność transformacji dodano pomocnicze znaczniki dla wierzchołków dla grafu z danych wejściowych)

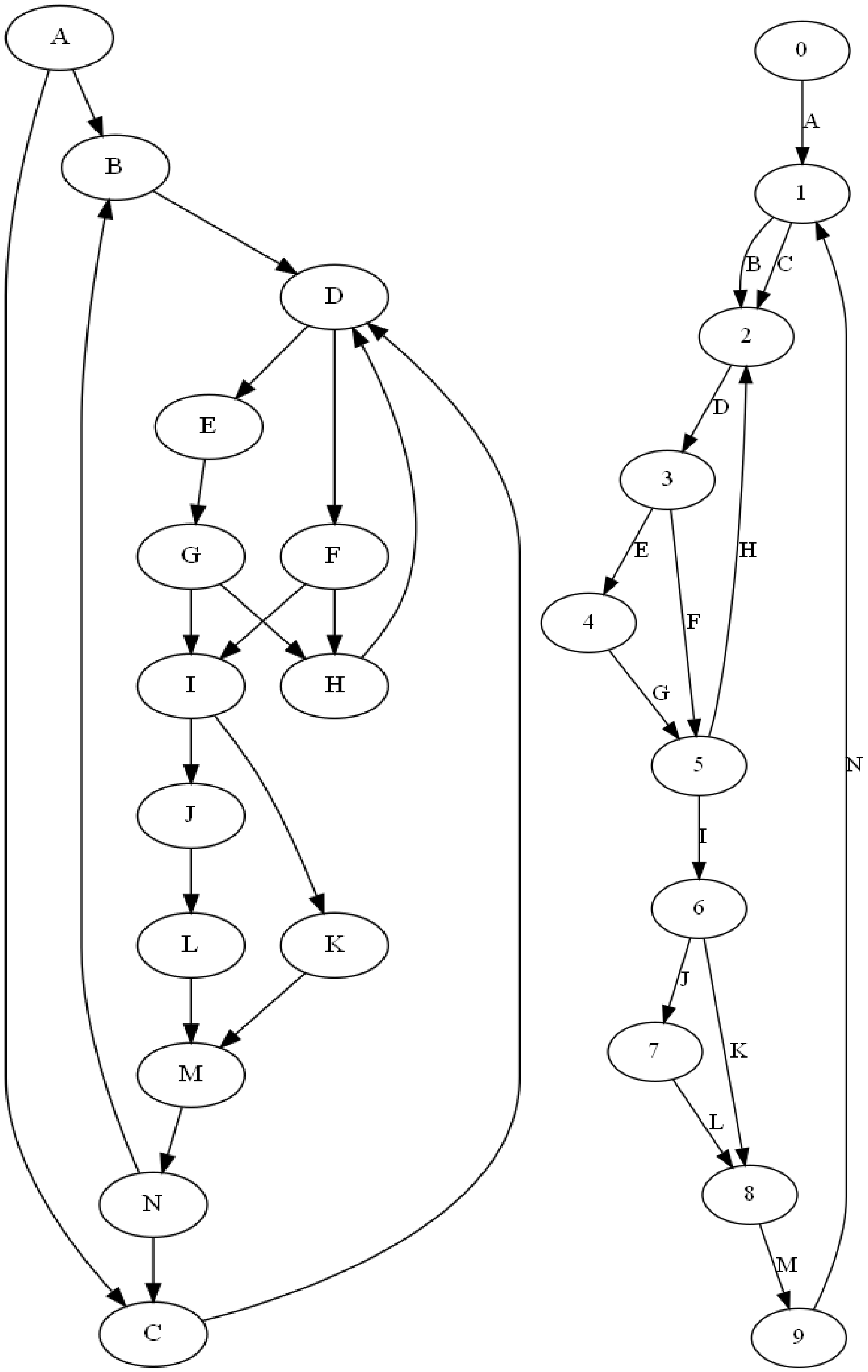
Przykłady grafów:

Graf wejściowy po lewej stronie, wygenerowany oryginalny po lewej

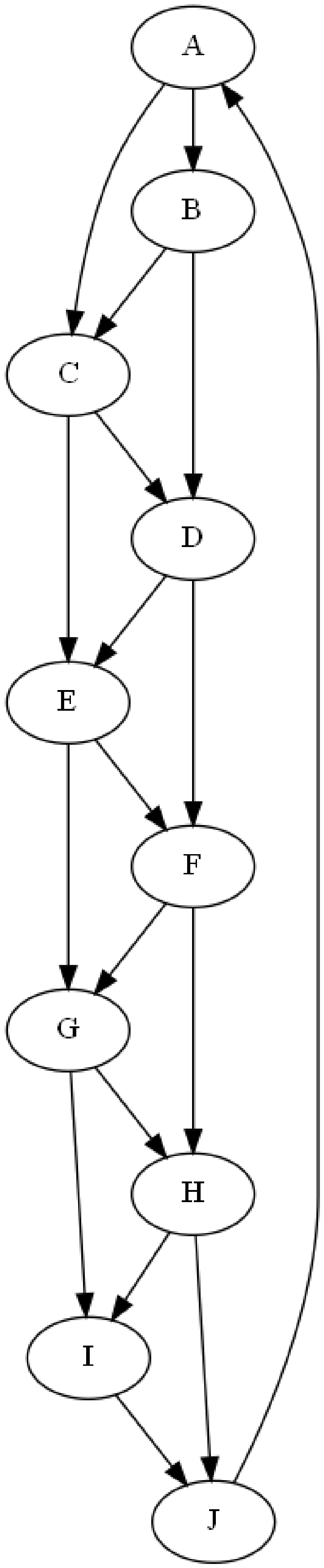
Przypadek nr 1: Graf sprzężony oraz liniowy



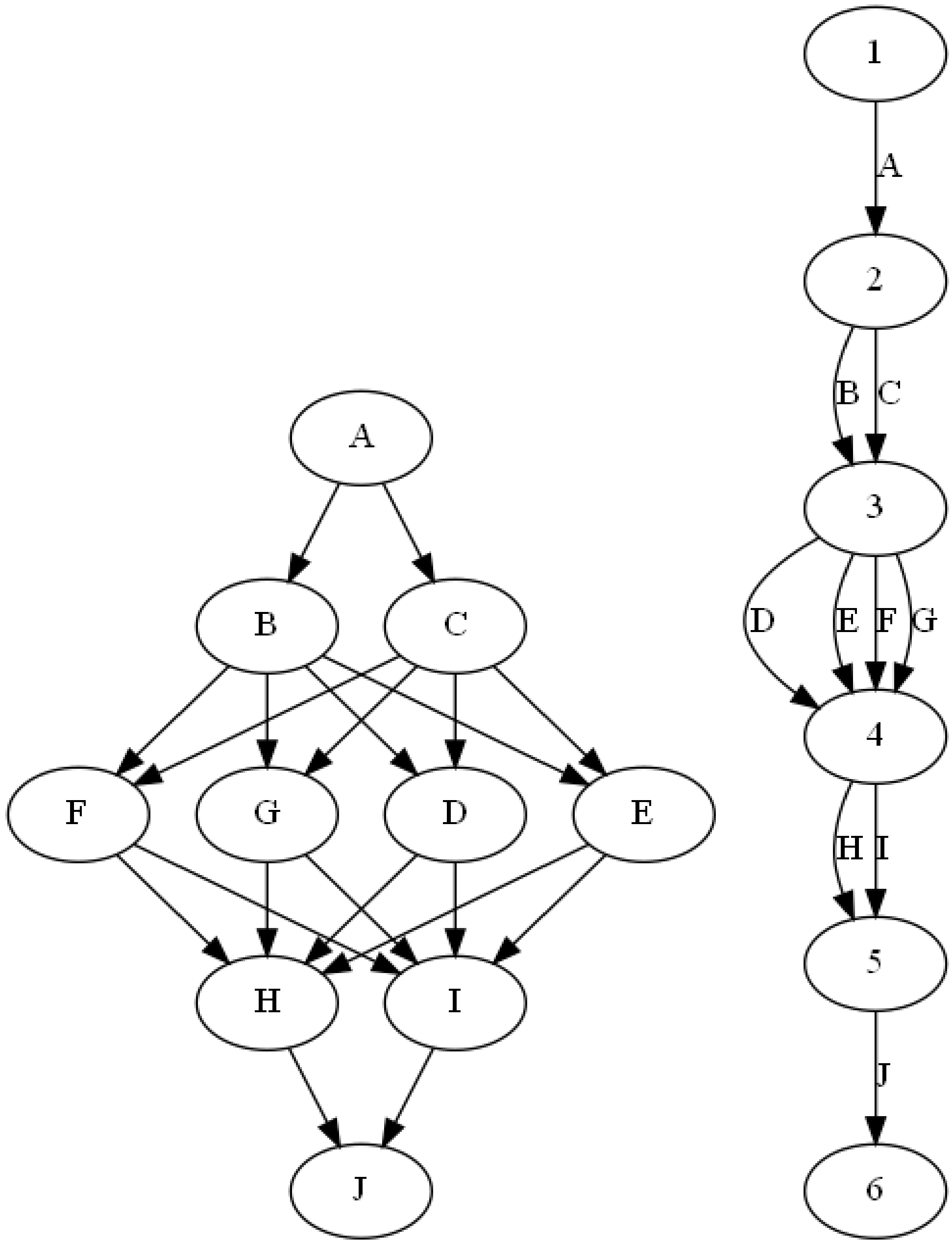
Przypadek nr 2: Graf sprzężony oraz liniowy



Przypadek nr 3: Graf nie jest sprzężony, brak grafu oryginalnego



Przypadek nr 4: Graf sprzężony, nieliniowy



1) Wnioski:

Sprawdzanie czy graf jest sprzężony lub liniowy należy do klasy problemów NP-zupełnych. Czas wykonywania rośnie wielomianowo w zależności od ilość wierzchołków. Podczas sprawdzania czy dowolny wygenerowany graf skierowany jest grafem sprzężonym oraz liniowym, sprawdzamy również czy jest możliwe sprawdzenie ścieżki Hamiltona w czasie wielomianowym. Jest to użyteczne, gdyż problem ścieżki Hamiltona jest problemem klasy NP-trudnej.