



Powyższy graf posłuży mi do omówienia złożoności metody topologicalSort. Wynik owego sortowania jest następujący : 6 5 3 4 2 1

W dużej mierze algorytm opiera się o DFS, dlatego też spodziewana złożoność powinna wynieść $O(|V| + |E|)$ czyli w tym przypadku 12 ($|V| = 6$, $|E| = 6$).

W głównej części znajduje się pętla, która przejdzie po wszystkich wierzchołkach i jeśli wierzchołek nie był odwiedzony to należy go odwiedzić za pomocą DFS-visit.

Zatem przy pierwszym przejściu pętli:

`i = 1; // patrze na wierzchołek 1`

odwiedzam wierzchołek, nie ma on żadnych połączeń stad nic więcej się nie wykona jak tylko włożenie na stos. Stąd 1 operacja. Podobnie z wierzchołkiem 2.

`i = 3;`

Wierzchołek 3 był nieodwiedzony zatem należy go odwiedzić. Przeglądamy połączenia, a więc idziemy w głąb do wierzchołka 4, ten też nie był odwiedzony, więc należy ponownie wykonać odwiedzanie. Patrzymy na kolejne połączenie z 2, ale ten wierzchołek był już odwiedzony.

Wierzchołek 4 nie ma więcej połączeń a zatem należy wracać. Wierzchołek 4 kładziony jest na stos i wracamy do wierzchołka 3. Ten też nie ma więcej połączeń więc kładziemy go na stos. Wracamy do głównej pętli.

Jak dotąd wykonaliśmy 6 operacji. 2 przy pierwszych dwóch wierzchołkach i 4 teraz , 2 na zejście do kolejnych wierzchołków i 2 na wrzucenie na stos.

W kolejnym etapie przeglądamy wierzchołek 4, ale był już odwiedzony więc przechodzimy do wierzchołka 5. Odwiedzając go sprawdzamy 2 połączenia wychodzące z niego czyli 2 operacje, bo kolejne wierzchołki które są z nim połączone zostały już odwiedzone. Wrzucamy go na stos czyli dodatkowa operacja.

Z wierzchołkiem 6 sytuacja jest podobna, czyli 3 operacje.

Łącznie: $2 + 6 + 6 = 12$, czyli tak jak przewidywaliśmy.

Złożoność pamięciowa wyniesie $O(|V|)$, bo wykorzystujemy stos do przetrzymywania wierzchołków już przetworzonych.