Laboratorium 4 - AiSD2

Dawid Kożykowski

Temat: Propagacja zmanipulowanych danych

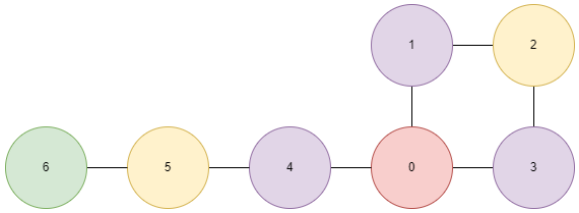
W pewnej firmie informatycznej infrastruktura systemu może być przedstawiona jako dwukierunkowy graf, w którym wierzchołki symbolizują serwisy, a krawędzie odzwierciedlają zależności między nimi. Relacja ta ma szczególny charakter – każdego ranka serwisy powiązane krawędzią wymieniają między sobą dane. Następnie, przez cały dzień, każdy serwis dokonuje obliczeń, **bazując na własnych wynikach z poprzedniego dnia oraz na nowych danych otrzymanych od swoich bezpośrednich sąsiadów**. Wyniki tych obliczeń zostają przesłane do sąsiednich serwisów kolejnego ranka.

Niestety, pewnego dnia wykryto, że w wyniku ataku hakerskiego dane jednego z serwisów zostały zmanipulowane. W konsekwencji zarówno wyniki generowane przez ten serwis, jak i wszystkie serwisy korzystające z tych danych, przestają być wiarygodne. Zarząd firmy chce ustalić, ile oraz które serwisy zostaną zainfekowane zmanipulowanymi danymi po upływie K dni. W tym celu zwrócił się do Ciebie z prośbą o pomoc. Czy podejmiesz się tego wyzwania?

Treść zadania:

Dany jest graf G reprezentujący infrastrukturę serwisów, liczby n, m, K, s oznaczające kolejno liczbę serwisów (serwisy numerowane są od 0), łączną liczbę zależności między serwisami, liczbę rozważanych dni oraz wierzchołek początkowy zaatakowany przez hakerów. Należy określić liczbę oraz podać listę serwisów, które po upływie K dni zostały zainfekowane wskutek propagacji zmanipulowanych danych.

Mechanizm propagacji infekcji:



Rysunek 1: Schemat propagacji zainfekowanych danych

Dla infrastruktury przedstawionej na rysunku 1, przyjmując, że $s = 0$ oraz $K = 3$, proces infekcji przebiega następująco:

- W dniu 1 zainfekowany zostaje serwis 0.
- W dniu 2, na podstawie danych z serwisu 0, infekcji ulegają serwisy 1, 3 oraz 4.
- W dniu 3, w wyniku propagacji infekcji z serwisu 4, zainfekowany zostaje serwis 5, natomiast serwis 2 zostaje zainfekowany na podstawie danych pochodzących z serwisów 1 lub 3.
- Serwis 6 pozostaje niezainfekowany w ciągu K dni.

W związku z powyższym, odpowiedzią dla tego przykładu jest 6 zainfekowanych serwisów: $0, 1, 2, 3, 4, 5$.

Etap 1. (0.5p)

W etapie pierwszym należy obliczyć liczbę i zwrócić listę zainfekowanych wierzchołków.

Algorytm musi działać w złożoności $\mathcal{O}(n+m)$ lub lepszej.

Etap 2. (1.0p)

W wyniku przeprowadzonej analizy problemu zauważono, że infekcji uległ więcej niż jeden serwis. Hakerzy zdołali zmanipulować dane p serwisów, których lista $s_1, s_2, \ldots, s_p$ zastąpiła parametr s z poprzedniego etapu.

Aby zapobiec pełnej propagacji zainfekowanych danych do wszystkich systemów, administratorzy postanowili wdrożyć środek zaradczy, który polega na wyłączaniu serwisów, zanim zostaną one zainfekowane wadliwymi danymi. Dzięki temu możliwe jest zatrzymanie dalszego rozprzestrzeniania się infekcji.

Do zbioru danych z poprzedniego etapu dodano tablicę *serviceTurnoffDay*, która dla każdego indeksu i przechowuje numer dnia, w którym i-ty serwis został wyłączony przez administratorów **przed** rozpoczęciem propagacji danych tego dnia. Jeśli wartość jest równa $K+1$, to serwis ten nigdy nie został wyłączony.

Twoje zadanie pozostaje niezmienne – nadal musisz analizować propagację danych, jednak z uwzględnieniem wyłączonych serwisów.

Algorytm musi działać w złożoności $\mathcal{O}(n+m)$ lub lepszej.

Etap 3. (1.0p)

Zarząd firmy zauważył, że wyłączanie serwisów może wiązać się z ogromnymi kosztami. Jednocześnie uznał pomysł administratorów za konieczny do wdrożenia. W celu minimalizacji strat zdecydowano się jednak na wprowadzenie pewnych zmian – serwisy będą wyłączane jedynie na określony czas, po czym zostaną ponownie uruchomione i wznowią swoje działanie.

Każdy serwis może zostać wyłączony tylko raz.

Do zbioru danych z poprzedniego etapu dodano tablicę *serviceTurnonDay*, która dla każdego indeksu i przechowuje numer dnia, na koniec którego i-ty serwis został ponownie włączony przez administratorów. Można założyć, że

$$\forall_i \ 1 \leqslant \mathit{serviceTurnoffDay}[i] \leqslant \mathit{serviceTurnonDay}[i] \leqslant K+1$$

Po ponownym włączeniu serwis kontynuuje propagację danych od stanu, w jakim znajdował się przed wyłączeniem. Oznacza to, że jeśli przed wyłączeniem był niezainfekowany, pozostaje niezainfekowany, a jeśli był zainfekowany, nadal pozostaje zainfekowany.

Algorytm musi działać w złożoności $\mathcal{O}(n+m+K)$ lub lepszej.

Uwagi:

- Serwisy numerowane są od 0
- Dni numerowoane są od 1 do K
- Warto rozważyć napisanie własnej wersji algorytmu przeszukiwania grafu, bazując na poznanych algorytmach.
- $1 \leqslant n, p, K < 10^6$, $0 \leqslant m < 10^6$
- W grafie G nie ma pętli (krawędzi do samego siebie) ani krawędzi wielokrotnych
- Graf G nie musi być spójny
- Początkowo zainfekowane serwisy zostają zaatakowane **po propagacji danych** 1. dnia, co oznacza, że propagują zainfekowane dane dopiero 2. dnia
- **Początkowo zainfekowane serwisy zostają zainfekowane po propagacji danych 1. dnia, nawet jeśli zostaną 1. dnia wyłączone. Wówczas zaczynają propagację zainfekowanych danych dopiero po ponownym włączeniu (etap 2. i 3.)**