

W ramach zajęć należało zaimplementować dowolny typ grafu za pomocą tablicy sąsiedztwa lub listy sąsiedztwa. Wybrano implementację za pomocą tablicy.

### **addEdge**

Gdy wierzchołki grafu między którymi ma zostać nawiązane połączenie istnieją, to złożoność czasowa dodania krawędzi wynosi  $O(1)$ . Jeśli jednak podane wierzchołki nie istnieją to wtedy, tablica zostaje powiększona do rozmiaru większego z nich, a poprzednie krawędzie zostają przekopiwane. Toteż Złożoność wyniesie  $O(V^2)$ , ponieważ należy przejść po wszystkich krawędziach między wierzchołkami i je przepisać, przy okazji odwiedzając miejsca gdzie krawędzi nie ma.

Złożoność Pamięciowa wyniesie  $(V^2)$ , ponieważ mamy do czynienia z tablicą 2D. W przykładzie podanym na zajęciach<sup>2</sup>, odpowiedniejszą reprezentacją byłaby reprezentacja za pomocą listy sąsiedztwa, ponieważ podany graf jest bardzo „dziurawy” przez co tracimy sporo nie używanej pamięci.

### **removeEdge**

Złożoność czasowa wyniesie  $O(1)$  ponieważ usunięcie krawędzi jest zmianą w odpowiednim miejscu wartości na 0. W tablicy sąsiedztwa dostęp do tego miejsca jest wykonywany w czasie  $O(1)$ .

### **IsEdge**

Złożoność czasowa tej operacji wyniesie  $O(1)$ , ponieważ wystarczy sprawdzić czy w danym miejscu, między wierzchołkami występuje połączenie.

---

1 Gdzie  $V$  to liczba wierzchołków

2 Plik 1000.csv