Jedná se o úlohu na grafy. Zadání po nás chce z hran obdržených z vstupu sestavit orientovaný graf („Každá hrana je orientovaná z křižovatky P do křižovatky Q.“) a odpovědět na dotazy, zda se dá dostat z křižovatky A do křižovatky B. Tj. jestli existuje cesta v tomto orientovaném grafu vedoucí z vrcholu A do vrcholu B. Protože nás nezajímá ani její průběh, ani délka, vystačíme si s úplně základním prohledáváním do šířky. Pro reprezentaci grafu jsem si vybral hash-mapu <vrchol sousede> kvůli rychlému vyhledávání vrcholů, když dostanu v dotazu počáteční vrchol, a kvůli nízké paměťové složitosti. Sousedy ukládám do spojového seznamu, protože stejně vždycky budu potřebovat projít všechny a proto nepotřebuji hash-set a je to i efektivní způsob pro ukládání dat – přidání nového elementu a rozšíření spojáku stojí O(1).

Časová složitost:

* vytvoření grafu ze vstupních dat: **O(E\*log(E))** – pomocí hash-mapy
* + dotazy:
  + počet dotazů **(Q)**
  + dotaz **q = O(E+V) – pomocí BFS**
* celková složitost: **O(E\*log(E) + q\*(E+V))**

Paměťová složisost:

* uložení grafu: každý vrchol může mít až **E** následovníků – pomocí hash-mapy <vrchol sousede> až **E^2**
* uložení dotazů: **Q**
* BFS: **V**
* Paměťová složitost bude tedy **O(E^2 + V)**, protože BFS probíhá pouze na jednom dotazu současně