

### 1. Algorytm Dijkstry

- a) uzupełnić kawałek kodu podany na kartce (trzeba było dopisać warunek  $d[v] > d[u] + w(u, v)$  )
- b) Jaką przykładową implementację kolejki można było by użyć w podanym kodzie, i jakie złożoności miały by 3 funkcje użyte w kodzie
- c) Złożoność obliczeniowa algorytmu i poszczególnych elementów kodu, za pomocą N-wierzchołków i M-krawędzi
- d) Złożoność obliczeniowa algorytmu za pomocą N-wierzchołków (trzeba było też napisać dlaczego tak się da)

### 2. Algorytm Bellmana-Forda

- a) Uzupełnić kawałek kodu (w głównej pętli trzeba było podać ile razy będzie się iterować (N-1))
- b) Dlaczego ma się tyle razy iterować
- c) złożoność obliczeniowa algorytmu
- d) dopisać na końcu kod który wykryje czy algorytm wpadł w ujemny cykl

### 3. 5 pytań prawda/fałsz

- a) Czy Algorytm Bellmana-Forda działa dla ujemnych cykli
- b) Czy Algorytm Dijkstry działa dla ujemnych wag
- c) Czy Bellmana-Forda działa dla ujemnych wag
- d) Czy algorytm Prima tworzy las rozpinający
- e) Czy istnieje tylko jedno Minimalne drzewo rozpinające w grafie

### 4. Algorytm Kruskala

- a) Podany jest graf z wagami i trzeba było wypisać krok po kroku wszystkie drzewa rozpinające jakie powstają
- b) Łączna waga tego drzewa po wykonaniu algorytmu
- c) Złożoność algorytmu
- d) Złożoność algorytmu Prima

### 5. Przechodnie domknięcia

- a) Przedstawić przechodnie domknięcie podanego grafu
- b) Jak można wykorzystać algorytm Floyd-Warshalla do wyznaczenia przechodniego domknięcia