Lista 2, zadanie 6 - Tomasz Woszczyński

Treść: Ułóż algorytm, który dla danego spójnego grafu G oraz krawędzi e sprawdza w czasie $O(n+m)$, czy krawędź e należy do jakiegoś minimalnego drzewa spinającego grafu G. Możesz założyć, że wszystkie wagi krawędzi są różne.

Rozwiązanie: Skorzystamy z cycle property dla MST mówiącego o tym, że dla każdego cyklu C, jeśli waga krawędzi e należącej do C jest większa od pozostałych wag w C, to ta krawędź **nie może** być w MST. Załóżmy więc, że e nie jest maksymalna na żadnym cyklu w grafie G i nie należy do MST. Rozpatrzmy więc przypadki:

1. krawędź e nie leży na żadnym cyklu: stąd oczywiste jest, że e należy do MST.
2. krawędź e leży na jakimś cyklu C: weźmy MST i dołóżmy do niego krawędź e - po tym kroku powstaje nam cykl, jako że e nie należało do MST. W tym drzewie rozpinającym musi istnieć jakaś krawędź maksymalna $\hat{e}$. Po usunięciu jej otrzymamy MST o mniejszej wadze, a więc dochodzimy do sprzeczności (MST to drzewo rozpinające o najmniejszej wadze).

Algorytm: Wiemy, że krawędź e łączy wierzchołki v, w. Zapamiętajmy wagę tej krawędzi i ją usuńmy. Z wierzchołka v puszczamy DFS, ale sprawdzamy jedynie wierzchołki połączone krawędziami o wagach nie większych od e. Jeśli dotrzemy do w, wypisujemy NIE, w przeciwnym wypadku wypisujemy TAK.

Wyjaśnienie: Jeśli nie dotarliśmy do w, to albo e jest mostem, albo e nie była maksymalna na żadnym cyklu, bo gdyby była, to przeszlibyśmy po każdej krawędzi tych cykli pomijając krawędź e. Złożoność czasową $O(n+m)$ osiągamy w bardzo łatwy sposób, gdyż wiemy, że DFS działa właśnie w takim czasie.