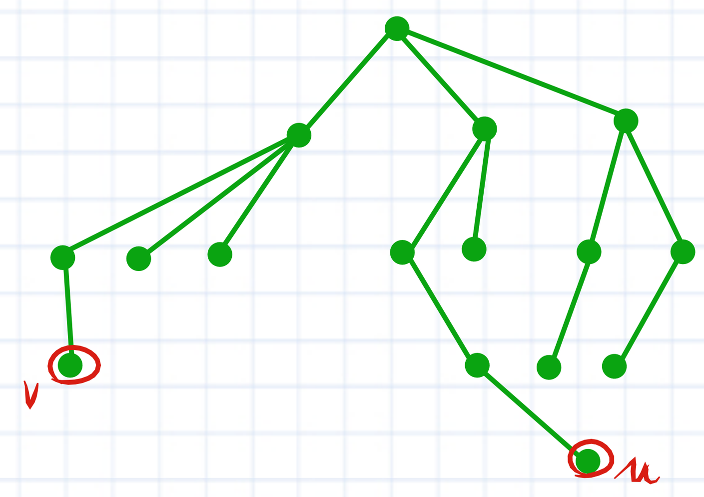
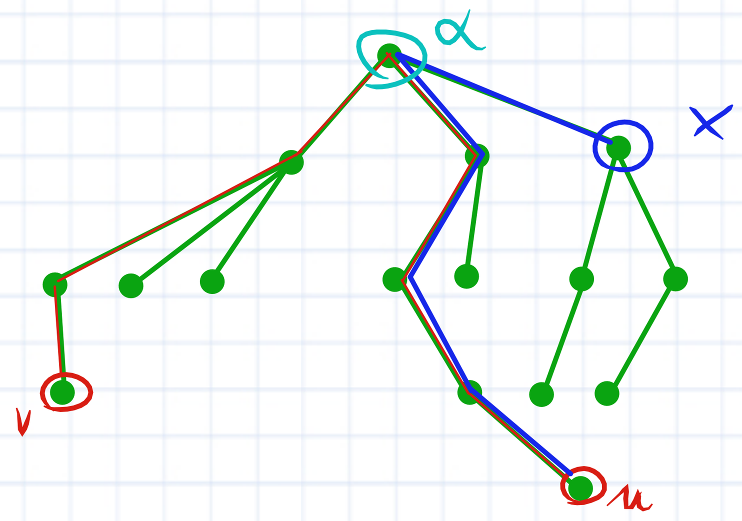
Weźmy dwa najbardziej oddalone od siebie wierzchołki, nazwijmy je v i u (na pewno są one liśćmi, bo gdyby tak nie było to ścieżkę można by wydłużyć o kolejny wierzchołek)



Wybieramy dowolny wierzchołek x jako wierzchołek startowy algorytmu (przypuszczalnie nie wiemy, gdzie jest korzeń drzewa) i odpalamy DFS-a, który zwraca nam najdalszy liść od v (gdyby to nie był liść to DFS poszedł by do kolejnego wierzchołka, gdyby to nie był najdalszy to w grafie musiałby być cykl)

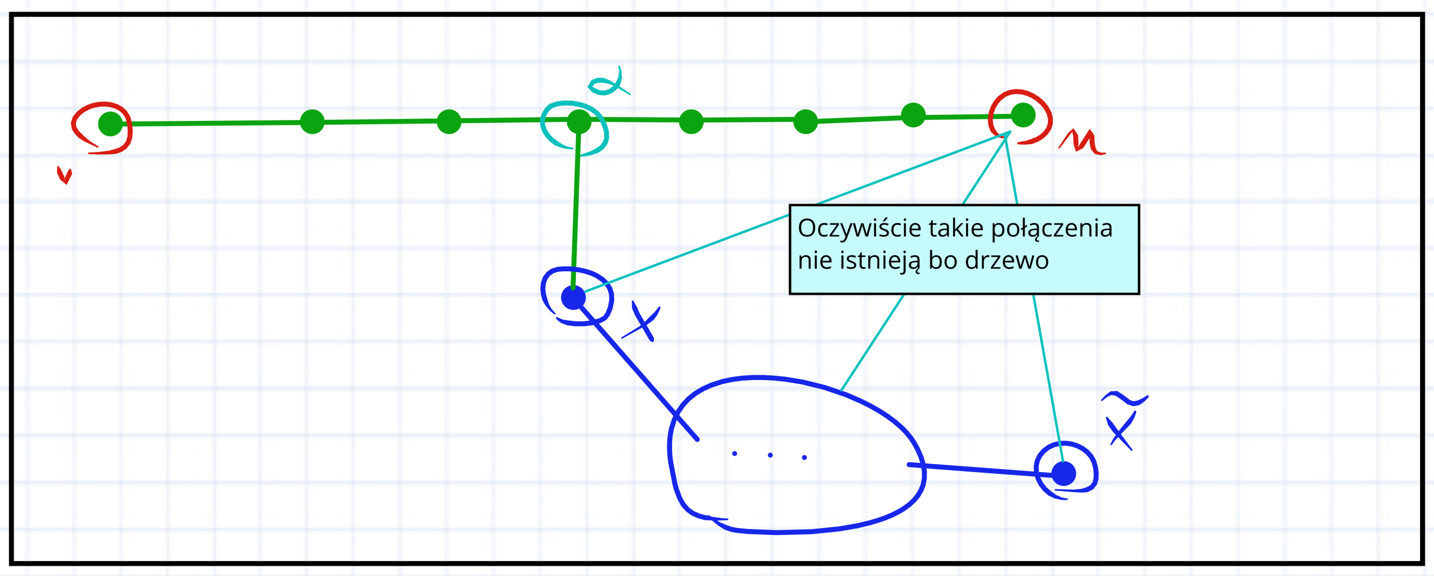


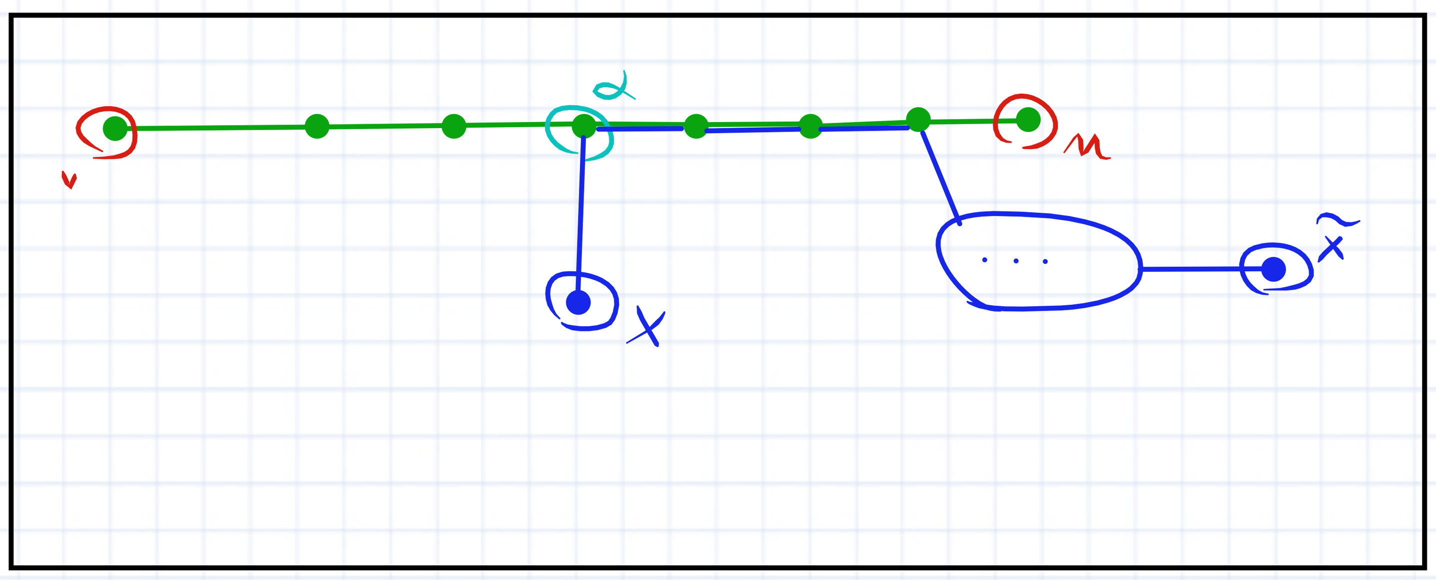
d(a, b) := odległość a do b

$$d(v,u) = d(u,\alpha) + d(v,\alpha)$$
$$d(x,u) = d(x,\alpha) + d(u,\alpha)$$
$$d(x,v) = d(x,\alpha) + d(v,\alpha)$$

Załóżmy, że istnieje $\tilde{x}$ taki, że $d(x,\tilde{x}) > \max(d(x,u), d(x,v))$

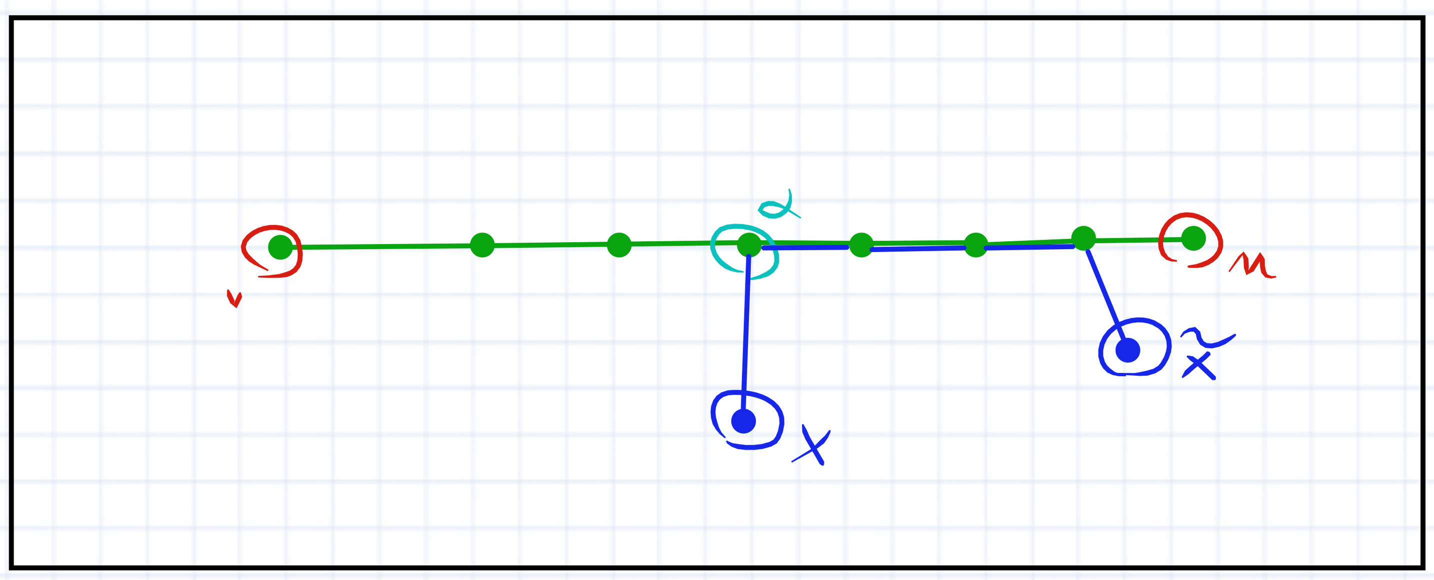
Przypadki:





to by znaczyło, że możemy wziąć ścieżkę o długości

$$d(x,\tilde{x}) + \max(d(x,u), d(x,v)) > d(v,u) \ ↯$$



Nieważne – $\tilde{x}$ jest po prostu nowym u, bo graf ma więcej niż jedną parę najbardziej odległych wierzchołków

Teraz wystarczy odpalić DFS-a dla tego liścia i znajdziemy liść od niego najbardziej oddalony, który razem z nim stanowi parę dwóch najdalszych wierzchołków.