

ASD2 Lab10 backtracking

Zadanie

Farmer Trotter ma świeże jajka. Parowóz Tomek ma musi rozwieźć jajka od farmera do miast i wsi na wyspie Sodor. W tym celu Gruby Zawiadowca przygotował specjalny, wyłożony marmurem i słomą, wagon do przewozu jajek. Niestety do rozwiezienia jest więcej jajek niż wynosi pojemność wagonu. To oznacza, że Tomek będzie musiał wielokrotnie wracać na farmę, aby uzupełnić jajka. Oprócz ograniczenia na pojemność wagonu, parowóz Tomek ma ograniczony zapas węgla i wody, które musi regularnie uzupełniać. Musi zatem zaplanować trasę, w której:

- każda ze stacji, która oczekuje na jajka zostanie odwiedzona dokładnie raz,
- od momentu wyjazdu z farmy do momentu do powrotu na farmę suma rozwiezionych jajek nie przekracza pojemności wagonu,
- odstęp pomiędzy odwiedzinami stacji, na których Tomek może uzupełnić węgiel i wodę nie przekracza zasięgu parowozu,
- trasa rozpoczyna się i kończy się na farmie

Potrzebny jest program, który zaplanuje trasę dla parowozu.

Na wejściu dane są następujące parametry:

- n -wierzchołkowy graf reprezentujący sieć kolejową na wyspie Sodor:
 - wierzchołek o indeksie 0 reprezentuje farmę,
 - pozostałe wierzchołki to stacje, do których trzeba dowieźć jajka,
 - wagi na krawędziach oznaczają czas przejazdu między stacjami,
- tablicę o długości n , w której znajdują się informacje o tym ile jajek trzeba zawieźć do odpowiedniej stacji. na zerowym indeksie (farma) zawsze jest 0, pozostałe wartości są dodatnie,
- liczbę naturalną c – liczbę jajek, która mieści się w wagonie,
- liczbę rzeczywistą dodatnią t – maksymalny czas jaki Tomek może przejechać bez uzupełniania węgla i wody,
- tablicę typu `bool`, która zawiera informację czy na odpowiedniej stacji da się uzupełnić wodę i węgiel. Stacja 0 nie musi być taką stacją!

Etap 1

Należy z użyciem algorytmu z nawrotami znaleźć jakąkolwiek trasę, która spełnia ograniczenia zadania, jeśli taka trasa istnieje. W tym etapie metoda zawsze ma ustawiony parametr `anySolution` na `true`.

Etap 2

Jeśli istnieje wiele możliwych tras, zwrócić tę, dla której czas przejazdu trasy jest najmniejszy. W tym etapie metoda zawsze ma ustawiony parametr `anySolution` na `false`.

Punktacja

- Etap 1
 - 1p zwrócenie informacji, czy trasa istnieje
 - 0,5p zwrócenie trasy
- Etap 2
 - 1p (całość etapu)