Lista 6

Zadanie 4

---

1 Problem

Jedziemy samochodem palącym 1 litr paliwa na 1 km. Samochód ma górne ograniczenie paliwa, które może przewieźć wynoszące W. Na drodze są porozstawiane stacje paliwowe, gdzie w_i to cena za 1 litr paliwa na stacji i. Należy znaleźć jak najtańszą drogę do stacji końcowej n.

2 Concept

Rozważamy drogę, po której jedzie samochód jako DAG (directed acyclic graph). Wierzchołkami w tym grafie są stacje. Krawędzie pomiędzy wierzchołkami tworzymy tylko wtedy kiedy pozwala na to zasięg samochodu. Jednakże, w takim modelu nie rozważamy sytuacji kiedy tankujemy więcej paliwa niż potrzebujemy do dojechania do następnej stacji, a może być sytuacja, że zatankowanie wcześniej większej ilości paliwa dałoby tańszą podróż. Musimy zatem rozszerzyć wierzchołki reprezentujące stacje o dodatkową informację — ilość pozostałego paliwa w baku samochodu.

Zatem rozbijamy wierzchołki reprezentujące poszczególne stacje na grupy wierzchołków oznaczanych przez parę (k, i) gdzie k to indeks stacji, a i to liczba litrów paliwa pozostałego w baku po dojechaniu do tej stacji. Co więcej, przez to, że rozważamy sytuacje, w których na niektórych stacjach tankujemy 0 litrów wówczas w grafie nie potrzebujemy rozważać krawędzi które pomijają stacje. Innymi słowy, jedynymi krawędziami są te pomiędzy wierzchołkami reprezentującymi sąsiednie stacje.

3 Rozwiązanie

Budujemy odpowiedni DAG $G = (V, E, f)$ gdzie V to wierzchołki grafu, E to krawędzie grafu a $f : V \times V \to \mathbb{R}$ to funkcja wag krawędzi. Mamy wierzchołki (k, i), gdzie k to indeks stacji, a i to liczba litrów paliwa „nadmiarowego" (ile litrów paliwa zostało po dojechaniu do $k$tej stacji) gdzie $i \in \{0, \ldots, W - 1\}$.

Ustalmy ciąg s_k określający kilometr stacji k (odległość stacji k od stacji startowej). Wówczas odległość między stacjami a, b wynosi $d_{a \to b} = s_b - s_a$.

Układ krawędzi i wierzchołków musi uwzględniać parametr W (zasięg samochodu). Zaczynamy od wierzchołka źródłowego $(1, 0)$ który reprezentuje stację, na której samochód rozpoczyna podróż. Dla pozostałych stacji budujemy wierzchołki (k, i) dla $i \in \{0, \ldots, W - d_{(k-1) \to k}\}$, $k \in \{2, \ldots, n\}$. Łączymy wierzchołki krawędziami uwzględniając zasięg oraz ile litrów paliwa „nadmiarowego" przewozi samochód. Czyli dla każdego wierzchołka (k, i) rozważamy tylko następną stację o wierzchołkach $(k+1, j)$ takich, że j jest z przedziału $[0, W - d_{k \to k+1}]$.

Funkcję wag krawędzi określamy jako $f\big((k, i), (l, j)\big) = w_k \cdot \big(d_{k \to l} - i + j\big)$. Wówczas za część przebytej drogi do następnej stacji możemy „zapłacić” paliwem kupionym na którejś z poprzednich stacji.

Stosujemy zmodyfikowany algorytm wyszukiwania najkrótszej drogi w DAG-u z zadania 1. Pomijamy $k = 1$ ponieważ jest to stacja źródłowa, do której nie potrzebujemy wiedzieć jak dojechać.

1: $d \leftarrow$ matrix $n \times W$ filled with ∞
2: $d[1][0] \leftarrow 0$
3: $r \leftarrow$ matrix $n \times W$ filled with [ ]
4: **for each** k in $\{2, \ldots, n\}$ **do**:
5:     **for each** i in $\{0, \ldots, W - d_{(k-1) \to k}\}$ **do**:
6:         **for each** j in $\{0, \ldots, W - d_{(k-2) \to (k-1)}\}$ **do**:
7:             **if** $d[k][i] > d[k-1][j] + f\big((k-1, j), (k, i)\big)$ **then**:
8:                 $d[k][i] \leftarrow d[k-1][j] + f\big((k-1, j), (k, i)\big)$
9:                 $r[k][i] \leftarrow \texttt{concat}\Big(r[k-1][j], \big[(k, i)\big]\Big)$
10:             **end if**
11:         **end for**
12:     **end for**
13: **end for**

Komentarz do linijek 5-6: W tych linijkach iteratory i oraz j są w okrojonym zakresie, który pomija niemożliwe wierzchołki (suma liczby litrów „nadmiarowych” i liczby litrów liczona za przejazd między stacjami przekracza W). Przy czym dla pierwszej iteracji k trzeba uważać na $k - 2$. Należy założyć, że dla $k = 2$ pętla z iteratorem j wykona tylko jedno przejście.

Złożoność obliczeniowa powyższego algorytmu wynosi $O(n \cdot W^2)$. Jest to górna granica, która określa przypadek, w którym stacje są rozmieszczone co kilometr przez co trzeba rozważać więcej przypadków zachowywania „nadmiarowego” paliwa.

Tablica $r[n][0]$ reprezentuje ciąg wierzchołków określających kolejne stacje i liczbę litrów pozostałego paliwa na tych stacjach podczas najtańszej podróży do końca drogi. Liczba $d[n][0]$ reprezentuje koszt tej podróży.

W celu określenia na jakiej stacji ile bierzemy litrów paliwa musimy odpowiednio przetworzyć ciąg wynikowy $r[n][0]$: niech R_p określa liczbę litrów paliwa pozostałą po dotarciu do stacji $p$tej (bierzemy stacje tylko z tego ciągu wynikowego z zachowaniem oryginalnych indeksów stacji). Ciągiem określającym nasz ostateczny wynik będzie S_n, gdzie S_p to liczba litrów paliwa, którą samochód musi zatankować na stacji $p$tej. Element tego ciągu obliczamy w następujący sposób: $S_p = d_{p \to p+1} + R_{p+1} - R_p$ przy czym $S_n = 0$.

Powyższy algorytm zawsze nam zwróci najtańszą drogę do celu, ponieważ „promuje” te stacje na których są najniższe ceny za paliwo. Dzięki rozważaniu sytuacji, w której bierzemy na stacji z tanim paliwem więcej paliwa niż potrzeba do dojechania do następnej stacji, mamy pewność że maksymalnie wykorzystujemy te stacje, w których jest najtaniej.