Politechnika Wrocławska

Badania operacyjne i optymalizacja dyskretna

Sprawozdanie z wykonania zadania

Autorzy: Sebastian Żółkiewicz *259337* Piotr Kulczycki *259366*

Kod przedmiotu: W04ISA-SM0401G, Grupa: 2

Termin zajęć: czwartek 11:15 - 13:00

Politechnika
Wrocławska

Spis treści

1 Numer ćwiczenia

Wykonano zadanie MinPath - algorytm Bellmana-Forda oraz Dijkstry.

2 Termin oddania + okres spóźnienia

Termin oddania zadania 21.11.2024.

3 Algorytm Dijkstry

Algorytm Dijkstry jest jednym z najpopularniejszych algorytmów stosowanych do znajdowania najkrótszej ścieżki w grafie, którego wagi krawędzi są nieujemne. Został opracowany przez holenderskiego informatyka Edsgera Dijkstrę w 1956 roku. Algorytm działa na grafach skierowanych i nieskierowanych, przeszukując wierzchołki w sposób zachłanny. Jego celem jest znalezienie najkrótszej drogi od jednego, zadanego wierzchołka początkowego do wszystkich innych wierzchołków w grafie.

3.1 Działanie

1. Wierzchołek źródłowy oznaczamy przez s, a wagę krawędzi pomiędzy wierzchołkami i i j jako $w(i, j)$.
2. Na początku tworzona jest tablica d, przechowująca odległości od źródła s dla wszystkich wierzchołków grafu:
   - $d[s] = 0$
   - Dla pozostałych wierzchołków $d[v] = \infty$
3. Następnie tworzona jest kolejka priorytetowa Q, zawierająca wszystkie wierzchołki grafu. Priorytetem kolejki jest aktualnie obliczona odległość od wierzchołka źródłowego s.
4. Dopóki kolejka Q nie jest pusta:
   (a) Z kolejki usuwany jest wierzchołek u o najniższym priorytecie (najbliższy źródła spośród nierozważonych).
   (b) Dla każdego sąsiada v wierzchołka u wykonywana jest operacja relaksacji:
      - Jeśli $d[u] + w(u, v) < d[v]$ (czyli istnieje krótsza ścieżka do v przez u niż dotychczas znana), to $d[v]$ jest aktualizowane na $d[u] + w(u, v)$.
5. Po zakończeniu działania algorytmu tablica d zawiera najkrótsze odległości od źródła s do wszystkich wierzchołków w grafie.
6. Dodatkowo, można przechowywać tablicę poprzedników, w której dla każdego wierzchołka zapisany jest jego bezpośredni poprzednik na najkrótszej ścieżce. Dzięki temu możliwe jest odtworzenie pełnej ścieżki od źródła do dowolnego wierzchołka. Podczas każdej operacji relaksacji u staje się poprzednikiem v.

Złożoność czasowa algorytmu Dijkstry zależy od implementacji. W ramach laboratorium zaimplementowano algorytm wykorzystując kopiec. dzięki czemu osiągnięto złożoność $O(V + E \log V)$, gdzie V to liczba wierzchołków, a E to liczba krawędzi.

4 Algorytm Bellmana-Forda

Algorytm Bellmana-Forda również służy do znajdowania najkrótszych ścieżek w grafie, jednak działa także na grafach z ujemnymi wagami krawędzi. Algorytm ten został opracowany przez Richarda Bellmana i Lestera Forda. Jest bardziej ogólny od algorytmu Dijkstry, ale działa wolniej, ponieważ ma złożoność czasową $O(V \cdot E)$.

4.1 Działanie

1. Dla każdego wierzchołka v w $V[G]$ wykonaj:
   - $d[v] = \infty$
   - poprzednik[v] = niezdefiniowane
2. Ustaw $d[s] = 0$
3. Dla i od 1 do $|V[G]| - 1$ wykonaj:
   (a) Dla każdej krawędzi (u, v) w $E[G]$ wykonaj:
   - Jeśli $d[v] > d[u] + w(u, v)$, to:
     - $d[v] = d[u] + w(u, v)$
     - poprzednik[v] = u

Algorytm Bellmana-Forda jest szczególnie przydatny w sytuacjach, gdy występują krawędzie o ujemnej wadze, czego algorytm Dijkstry nie obsługuje poprawnie.

5 Dane wejściowe

Dane dostarczone przez prowadzącego reprezentują graf skierowany z dodatnimi wagami na krawędziach.

- Pierwsza liczba N określa liczbę wierzchołków grafu, w tym przypadku $N = 5$.
- Kolejne N wierszy zawiera po N liczb, które przedstawiają wagę krawędzi od węzła A (określonego przez numer wiersza) do węzła B (określonego przez numer kolumny).
- Waga wynosząca 0 oznacza brak połączenia między wierzchołkami A i B.

Przykładowa macierz wag wygląda następująco:

$$\begin{matrix} 5 & 0 & 0 & 0 & 8 \\ 0 & 0 & 0 & 7 & 0 \\ 0 & 3 & 0 & 6 & 0 \\ 8 & 0 & 0 & 0 & 8 \\ 0 & 9 & 0 & 6 & 0 \end{matrix}$$

6 Badania

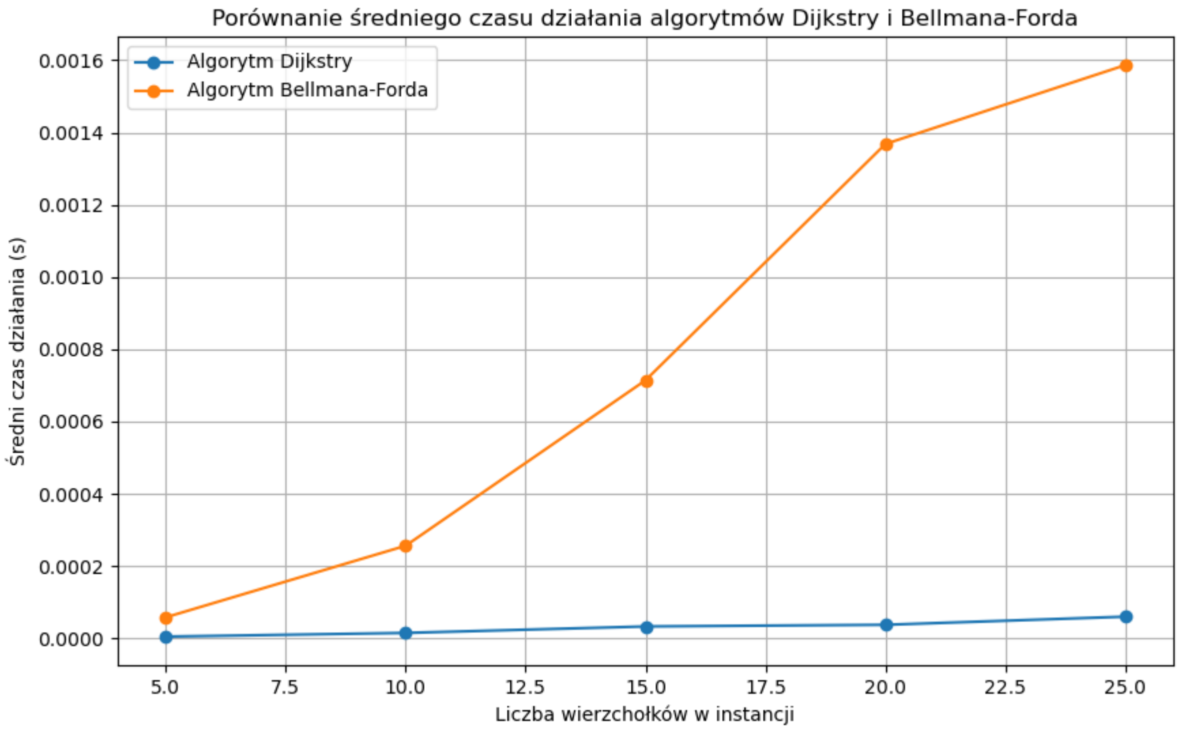
W ramach przeprowadzonych badań dokonano porównania dwóch algorytmów znajdowania najkrótszych ścieżek w grafach: algorytmu Bellmana-Forda oraz algorytmu Dijkstry. Celem tych badań było określenie ich złożoności czasowej.

6.1 Metodyka badań

- **Dane wejściowe:** Przeprowadzono testy na różnej wielkości grafach skierowanych z nieujemnymi wagami. Dane dostarczone przez prowadzącego kurs.
- **Kryteria porównawcze:** Czas działania algorytmów, zmierzony wielokrotnie.
- **Środowisko testowe:** Pomiary przeprowadzono na laptopie MB Pro M1 2021.

6.2 Wyniki

Na poniższym wykresie 1 przedstawiono średni czas trwania algorytmu w zależności od wielkości instancji.



Rysunek 1: Wykres średniego czasu działania algorytmów w zależności od wielkości instancji.

Średni czas wykonywania algorytmu Dijstry jest mniejszy od drugiego algorytmu. Dzięki zastosowaniu implementacji z wykorzystaniem kopca, złożoność obliczeniowa wyniosła $O(E \log V)$. Za to złożoność algorytmu Bellmana-Forda wynosi $O(|V| \cdot |E|)$. Zatem uzyskane wyniki pokrywają się z wiedzą teoretyczną.

7 Wnioski

Z przeprowadzonych badań wynika, że wybór algorytmu do wyznaczania najkrótszych ścieżek w grafie powinien być uzależniony od specyfiki badanego problemu:

- **Algorytm Dijkstry** jest bardziej efektywny w przypadku grafów z dodatnimi wagami krawędzi. Dzięki zastosowaniu struktury kolejki priorytetowej jego złożoność wynosi $O(E \log V)$, co czyni go szybszym od algorytmu Bellmana-Forda w większości praktycznych zastosowań.
- **Algorytm Bellmana-Forda** wykazuje większą wszechstronność, ponieważ jest w stanie obsługiwać grafy z krawędziami o ujemnych wagach oraz identyfikować cykle o ujemnej sumie wag. Jego złożoność czasowa wynosi $O(V \cdot E)$, co sprawia, że działa wolniej niż algorytm Dijkstry, zwłaszcza dla dużych i gęstych grafów.
- Dla grafów z dodatnimi wagami algorytm Dijkstry jest preferowanym wyborem ze względu na wyższą wydajność. W przypadku grafów z potencjalnie ujemnymi wagami krawędzi należy stosować algorytm Bellmana-Forda.
- W przypadku obecności cykli o ujemnej sumie wag, algorytm Bellmana-Forda umożliwia ich wykrycie.

8 Źródła

- Materiały z wykładu i zajęć.
- Materiały dostarczone przez prowadzącego: http://mariusz.makuchowski.staff.iiar.pwr.wroc.pl
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. Introduction to Algorithms, Third Edition (3rd. ed.). The MIT Press.