Algorytmy i struktury danych - projekt
Zwiedzanie Muzeów

Ewelina Preś, Franciszek Saliński

Grudzień 2024

1 Treść zadania

Bitowice to miasto o bardzo wielu atrakcjach turystycznych. Jest tam k bardzo ciekawych muzeów. Sieć komunikacyjna w Bitowicach obejmuje n przystanków autobusowych, które łączy m ulic. Przystanek nr 1 to dworzec kolejowy. Parę przystanków p i q może łączyć co najwyżej jedna ulica, a autobusy kursują w obie strony. Znany jest czas przejazdu autobusem pomiędzy przystankami. Przy każdym z k muzeów jest dokładnie jeden przystanek autobusowy. W najbliższy weekend Bajtazar postanowił przyjechać wczesnym rankiem do Bajtowic i zwiedzić wszystkie muzea. Swoją trasę rozpoczyna i kończy na przystanku autobusowym przy dworcu kolejowym. Bajtazar postanowił, że będzie zwedzał muzea w kolejności rosnącej numerów przystanków, przy których znajdują się te muzea. Niestety, czas ma ograniczony, więc chce tak określic trasę przejazdu autobusami, by łączny czas przejazdu był minimalny.

Wejście:
Należy wylosować liczbę n (liczba przystanków autobusowych), liczbę m (liczba połączeń między przystankami) oraz liczbę k muzeów ($k <= n$). Następnie, dla każdej pary przystanków, które łączy trasa autobusowa, losowo podać czas przejazdu.

Wyjście:
Program podaje minimalny czas przejazdu aububusami na trasie wycieczki Bajtazara.

2 Rozwiązanie

Problem zwiedzania muzeów przez Bajtazara zamodelowaliśmy za pomocą grafu ważonego, gdzie wierzchołki to przystanki, krawędzie to ulice, a muzea znajdują się w pewnych wyróżnionych wierzchołkach. Do rozwiązania problemu wykorzystaliśmy algorytm Dijkstry, który pozwalał nam na znalezienie najkrótszej ścieżki od pewnego wierzchołka, w którym się aktualnie znajdujemy, do wierzchołka, do którego chcemy się dostać. W funkcji route odtwarzamy najkrótszą ścieżkę wyznaczoną przez algorytm Dijkstry i obliczamy czas przejazdu.

2.1 Algorytm Dijkstry

Algorithm 1 Algorytm Dijkstry

procedure DIJKSTRA(V, E, s)
forall $v \in V$ **do**
$D[v]$.dist $\leftarrow +\infty$;
$D[v]$.prev $\leftarrow$ null;
od;
$D[s]$.dist $\leftarrow 0$;
$S \leftarrow \emptyset$;
while $S \neq V$ **do**
$mindist \leftarrow +\infty$;
forall $w \in V \setminus S$ **do**
if $D[w]$.dist $< mindist$ **then**
$v \leftarrow w$;
$mindist \leftarrow D[w]$.dist;
fi;
od;
$S \leftarrow S \cup \{v\}$;
forall $w \in V - S$ **do**
if $D[w]$.dist $> mindist + l(v, w)$ **then**
$D[w]$.dist $:= mindist + l(v, w)$;
$D[w]$.prev $:= v$;
fi;
od;
od;
end procedure

Jak wiemy, algorytm **Dijkstry** w implementacji opartej na macierzy sąsiedztwa ma złożoność czasową $\mathcal{O}(n^2)$, gdzie n oznacza liczbę wierzchołków grafu.

2.2 Algorytm rozwiązujący zadanie

Algorithm 2 Funkcja ROUTE - wyznaczanie optymalnej trasy

```
procedure ROUTE((M, G))                                ▷ M - lista muzeów, G - graf
    sort M;                                            ▷ Sortowanie listy muzeów
    current_stop ← 1;
    time ← 0;
    route ← [1];
    forall i ∈ M do
        distances ← DIJKSTRA(current_stop, G);
        time ← time + distances[i].dist;
        bus_stop ← i;
        subroute ← [bus_stop];
        while bus_stop ≠ current_stop do
            bus_stop ← distances[bus_stop].previous;
            append bus_stop to subroute;
        od;
        route ← route + reverse(subroute)[1 :];
        current_stop ← i;
    od;
    distances ← DIJKSTRA(current_stop, G);
    time ← time + distances[1].dist;
    bus_stop ← 1;
    subroute ← [bus_stop];
    while bus_stop ≠ current_stop do
        bus_stop ← distances[bus_stop].previous;
        append bus_stop to subroute;
    od;
    route ← route + reverse(subroute)[1 :];
    return (time, route);
end procedure
```

2.3 Złożoność obliczeniowa naszego programu

Analiza złożoności algorytmu **ROUTE**:

- Algorytm najpierw sortuje listę muzeów M o długości k, co ma złożoność $\mathcal{O}(k \log k)$.
- Następnie dla każdego muzeum wykonuje się algorytm Dijkstry, którego złożoność wynosi $\mathcal{O}(n^2)$, gdzie n to liczba wierzchołków grafu.
- Algorytm Dijkstry jest wywoływany $k+1$ razy (dla każdego muzeum i powrotu do punktu początkowego).
- Dla każdej trasy algorytm rekonstruuje ścieżkę o długości co najwyżej n, co ma złożoność liniową $\mathcal{O}(n)$.

Podsumowując, złożoność czasowa algorytmu **ROUTE** wynosi:

$$\mathcal{O}(k \log k) + (k+1) \cdot \mathcal{O}(n^2 + n),$$

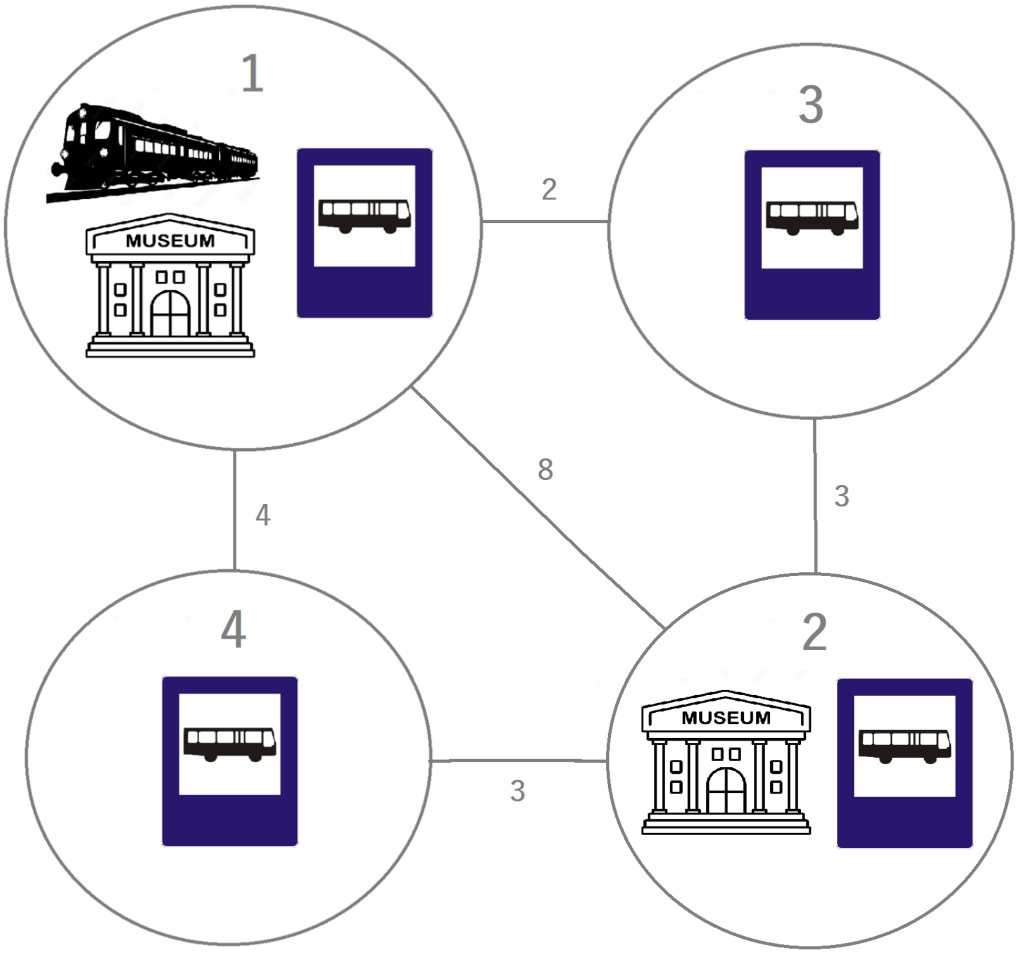
co w przybliżeniu daje $\mathcal{O}(kn^2)$.

3 Przykłady

Przykład 1

W pierwszym przykładzie mamy w Bitowicach liczbę muzeów równą 2 i liczbę przystanków równą 4. Przystanki połączone są 5 ulicami. Układ dróg i przystanków ukazuje rysunek 1.

Według naszego programu najbardziej optymalną droga zwiedzania muzeów to 1 $->$ 3 $->$ 2 $->$ 3 $->$ 1.
Czas tego przejazdu wynosi 10.



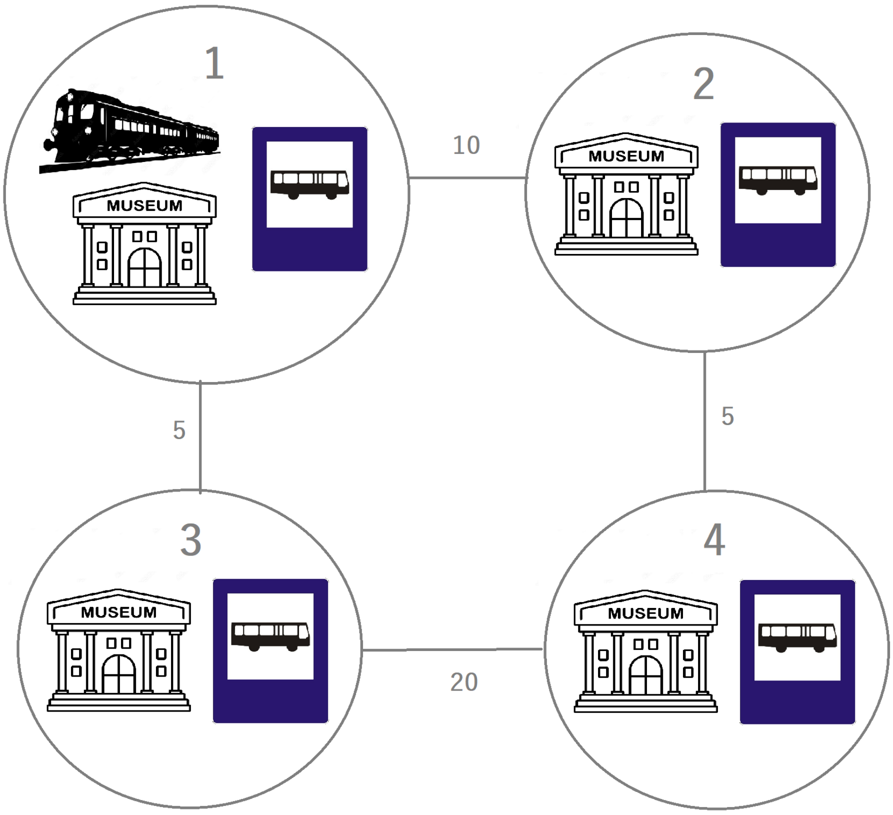
Rysunek 1:

Rzeczywiście, jeśli zwiedzamy muzea w kolejności rosnącej numerów przystanków, to zaczynając od razu w muzeum przy dworcu, najkrótsza droga dotarcia do muzeum przy przystanku 2 to droga przez przystanek 3. Najkrótszą drogą powrotną jest oczywiście ta sama droga, przez przystanek 3. Łatwo policzyć, że jest to faktycznie najkrótsza trasa, a czas przejazdu wynosi 10.

Przykład 2

W drugim przykładzie mamy 4 przystanki, połączone czterema drogami i przy każdym przystanku jest muzeum (rysunek 2).

Trasa wskazana przez nasz program to 1 − > 2 − > 1 − > 3 − > 4 − > 2 − > 1. Czas tego przejazdu wynosi 60.



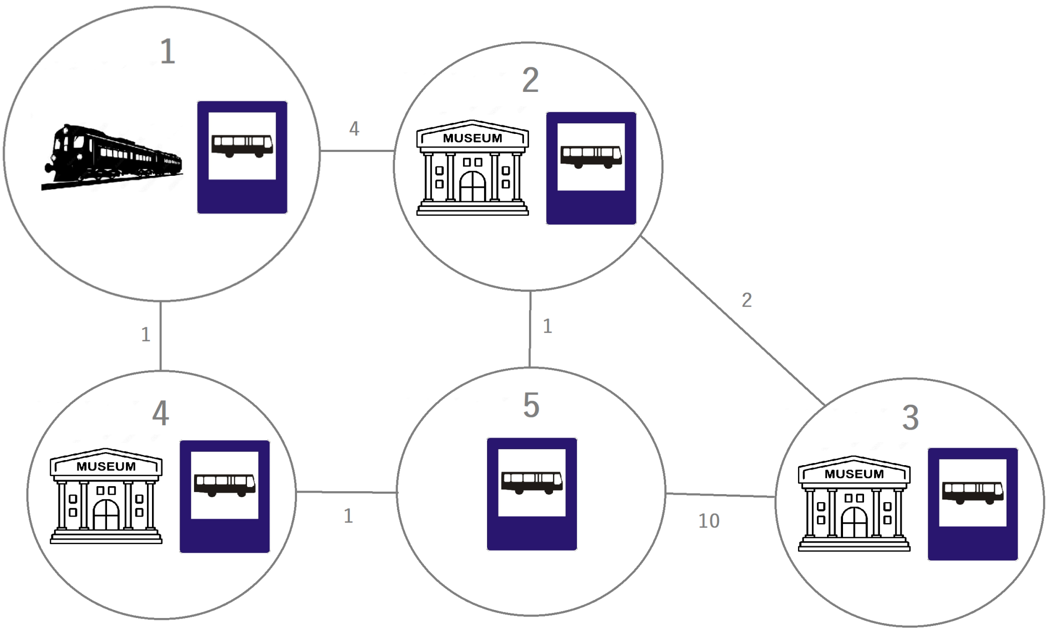
Rysunek 2:

Przeanalizujmy w jaki sposób przebiega trasa. Musimy odwiedzić muzea w kolejności 1, 2, 3, 4 znajdujących się przy nich przystanków. Startujemy z przystanku numer 1, a najkrótsza trasa do przystanku 2 to trasa bezpośrednia. Aby z przystanku 2 dotrzeć do przystanku 3, bardziej opłaca nam się wybrać trasę przez przystanek 1 (z czasem przejazdu 15) niż przez przystanek 4 (z czasem przejazdu 25). Z przystanku 3 do przystanku 4 można przejechać na dwa równie optymalne sposoby, które mają czas przejazdu 20. Nasz program w tym przypadku wybrał trasę bezpośrednią z 3 do 4. Powrót z przystanku 4 do przystanku 1 odbywa się przez przystanek 2, jest to najkrótsza trasa. Podliczając wszystkie wagi krawędzi, którymi przechodziliśmy, widzimy, że faktycznie czas przejazdu wynosi 60.

Przykład 3

W przykładzie 3 mamy graf składający się z 5 wierzchołków, połączonych 6 krawędziami. Przy trzech z tych przystanków znajdują się muzea (rysunek 3).

Trasa wyznaczona przez nasz program: 1 − > 4 − > 5 − > 2 − > 3 − > 2 − > 5 − > 4 − > 1
Czas przejazdu: 10



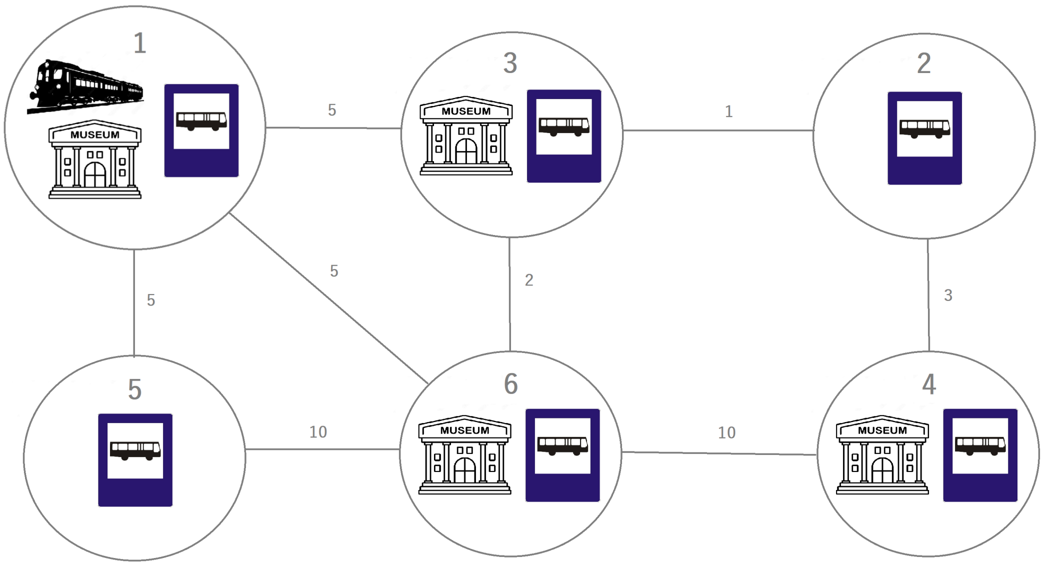
Rysunek 3:

Przeanalizujmy trasę przejazdu. Zaczynamy na przystanku 1 i chcemy udać się na przystanek 2, przy którym jest muzeum. Najkrótszą drogą jest droga przez przystanek 4 i 5 (z czasem przejazdu 3). Przejazd z przystanku 2 na 3 najszybciej odbywa się trasą bezpośrednią (czas przejazdu: 2). Następnie z przystanku 3 na przystanek 4 wracamy taką samą trasą, jaką jechaliśmy na początku: przez przystanek 2 i 5. Z przystanku 4 wracamy bezpośrednim połączeniem na przystanek 1. Czas takiego przejazdu faktycznie wynosi 10.

Przykład 4

W przykładzie 4 w Bitowicach mamy 6 przystanków, przy czym przy 4 z nich znajduje się muzeum. Przystanki połączone są 8 krawędziami (rysunek 4).

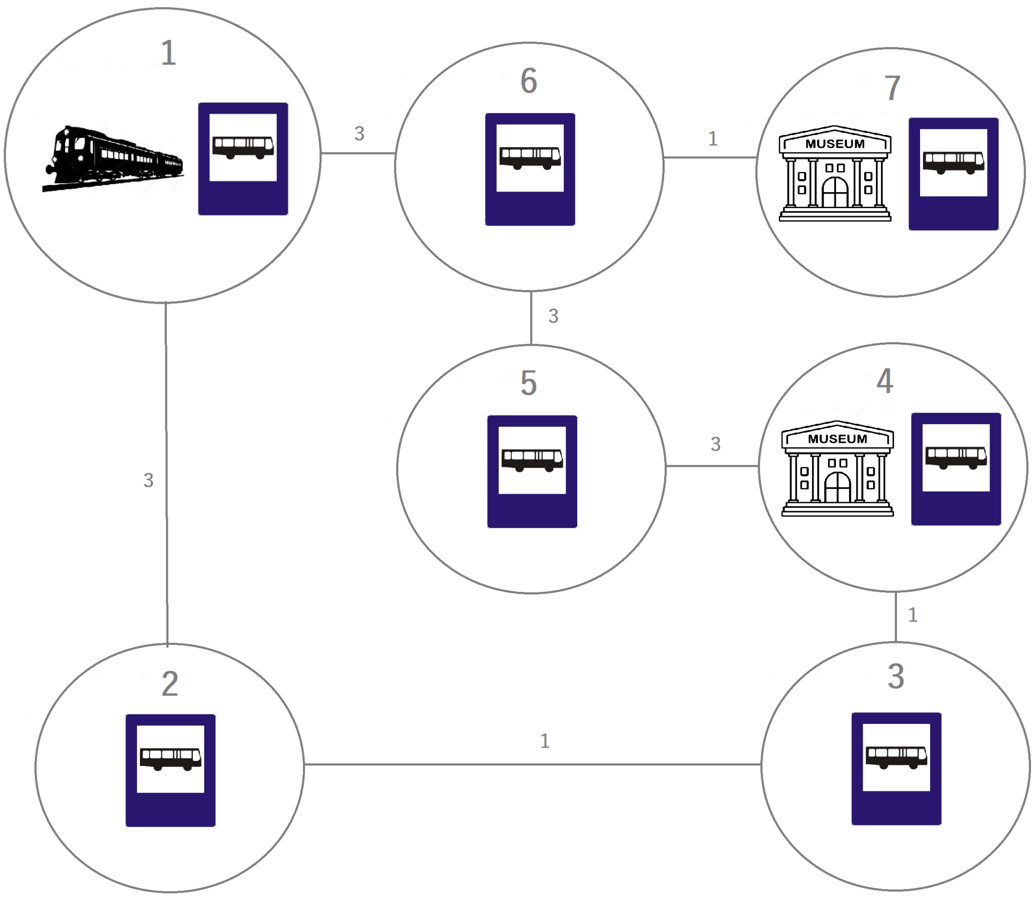
Trasa wyznaczona przez nasz program: 1 − > 3 − > 2 − > 4 − > 2 − > 3 − > 6 − > 1
Czas przejazdu: 20



Rysunek 4:

Istotnie, najktrótszą drogą z przystanku 1 do przystanku 3 jest trasa bezpośrednia (z czasem przejazdu 5). Następnie chcąc przejechać z przystanku 3 na przystanek 4, najoptymalniej jest wybrać drogę przez przystanek 2 (z łącznym czasem przejazdu 4). Aby z przystanku 4 dostać się na przystanek 6, najlepiej jest wybrać drogę przez przystanek 2 i 3 (z łącznym czasem przejazdu 6). Najkrótszą drogą z przystanku 6 z powrotem na dworzec jest bezpośrednie połączenie z czasem przejazdu 5. Całkowity czas przejazdu faktycznie wynosi 20.

Przykład 5



Rysunek 5:

W ostatnim przykładzie mamy w Bitowicach 7 przystanków autobusowych, ale tylko 2 muzea. Przystanki połączone są 7 krawędziami (rysunek 5).

Trasa wyznaczona przez nasz program: $1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 6 -> 1$
Czas przejazdu: 16

Łatwo zauważyć, że najkrótszą drogą z przystanku 1 do przystanku 4 jest faktycznie droga przez przystanek 2 i 3 (czas przejazdu: 5). Z przystanku 4 na przystanek 7 najszybciej jest dotrzeć przez przystanki 5 i 6 (czas przejazdu: 7). Z przystanku 7 dotrzemy szybko na dworzec kolejowy przejeżdżając przez przystanek 6 (czas przejazdu: 4). A zatem łączny czas przejazdu faktycznie wynosi 16.