

## **Zespół nr: 1**

### **Skład zespołu wraz z funkcjami:**

#### **Paweł Szczepankiewicz:**

**Koordynator, Programista, Tester, Autor dokumentacji**

#### **Kamil Nalewajski:**

**Programista, Tester, Strona graficzna**

#### **Konrad Zdziarski:**

**Programista, Tester, Strona graficzna**

Problem	Specyfikacja problemu (dane i wyniki)	Do jakich treści w zadaniu odnosi się algorytm	Zastosowane struktury danych	Informacje o zastosowanym algorytmie
1	<p><b>Dane:</b></p> <p>Pole – identyfikator, współrzędne (x, y), ilość jęczmienia</p> <p>Browar – identyfikator, współrzędne (x, y), pojemność produkcyjna</p> <p>Karczma – identyfikator, współrzędne (x, y), zapotrzebowanie</p> <p>Droga – połączenie między obiektami (źródło, cel), produkt przewożony, przepustowość, koszt naprawy</p> <p><i>(Dodatkowo w strukturze znajduje się WynikTrie, ale to element przyszłych problemów – np. Problem 5)</i></p> <p><b>Wynik:</b></p> <p>Reprezentacja danych w postaci struktur w programie</p> <p>Przygotowanie danych do przetwarzania przez algorytmy w kolejnych problemach</p>	<p>“Zaproponuj jak informacje o polach, browarach, karczmach i drogach reprezentować w komputerze“</p>	<p>Struct</p> <p>Vector</p> <p>Map</p>	<p>Brak zastosowanych algorytmów, użyto struktur.</p>

2	<p><b>Dane:</b></p> <p>Graf przepływu zbudowany z:</p> <p>Pol (Pole), browarów (Browar) i karczm (Karczma) jako wierzchołków</p> <p>Dróg (Droga) jako krawędzi z przepustowościami</p> <p>Reprezentacja jako macierz sąsiedztwa</p> <p><b>Wynik:</b></p> <p>Maksymalna ilość piwa możliwa do dostarczenia do karczm</p>	<p>“Opracuj sposób znalezienia maksymalnej ilości piwa, która można dostarczyć do karczm w Shire”</p>	<p>Macierz sąsiedztwa</p> <p>Graf skierowany z wagami</p> <p>Kolejka</p> <p>Wektory</p>	<p>Algorytm <b>Edmondsa-Karpa</b> [maksymalny przepływ <math>O(V \cdot E^2)</math>]</p> <p>BFS (w każdej iteracji przepływu)</p>
---	---	---	---	--

3	<p><b>Dane:</b></p> <p>Graf skierowany z krawędziami zawierającymi:</p> <p>Pojemność (przepustowość)</p> <p>Koszt przesyłu jednostki towaru</p> <p>Aktualny przepływ</p> <p>Struktura grafu: lista sąsiedztwa (vector&lt;vector&lt;Krawedz&gt;&gt;)</p> <p><b>Wynik:</b></p> <p>Maksymalny przepływ (max_flow)</p> <p>Minimalny koszt realizacji tego przepływu (min_cost)</p>	<p>“Zmodyfikujcie swoje rozwiązanie tak, żeby przy zachowaniu ilości przewożonego towaru, koszt naprawy dróg, po których poruszają się transporty był możliwie najmniejszy”</p>	<p>Lista sąsiedztwa – dynamiczne przechowywanie grafu</p> <p>Kolejka priorytetowa do optymalizacji wyszukiwania ścieżek</p>	<p>Algorytm Successive Shortest Path SSP [<math>O(F \cdot (V + E \cdot \log V))</math>] – znajdowanie minimalnego kosztu przepływu</p> <p>Algorytm <b>Dijkstry</b> z potencjałami [<math>O(E \cdot \log V)</math> na iterację] – wyznaczanie najtańszych ścieżek</p>
---	--	---	---	--

4	<p><b>Dane:</b></p> <p>Współrzędne punktów granicznych każdej ćwiartki pola (Punkt)</p> <p>Ilość jęczmienia produkowanego na każdej ćwiartce (dane dodatkowe)</p> <p><b>Wynik:</b></p> <p>Współrzędne punktów granicznych każdej ćwiartki pola (Punkt)</p> <p>Ilość jęczmienia produkowanego na każdej ćwiartce (dane dodatkowe)</p>	<p>“Samwise kazał zebrać współrzędne punktów granicznych każdej ćwiartki (każda z ćwiartek okazała się być wielokątem wypukłym, rozłącznym z pozostałymi ćwiartkami). Wie też ile jęczmienia wyrasta na polu w poszczególnych ćwiartkach. Zmodyfikujcie swoje rozwiązanie uwzględniając te nowe informacje zebrane przez burmistrza Sama”</p>	<p>Struktura Punkt (x, y)</p> <p>Sortowanie punktów wg x, następnie y</p> <p>Iloczyn wektorowy do określania orientacji punktów</p> <p>Wzór na pole wielokąta ("shoelace formula")</p>	<p>Algorytm <b>Grahama</b> [wypukła otoczka <math>O(n \cdot \log n)</math>]</p>
---	--	---	--	---

5	<p><b>Dane:</b></p> <p>Plik tekstowy zawierający treść do przeszukania.</p> <p>Lista poszukiwanych słów (wzorców)</p> <p><b>Wynik:</b></p> <p>Pozycje wystąpień poszukiwanych słów w tekście</p>	<p>“Burmistrz Marzy o tym, żeby szybko wyszukiwać w tych rozwiązaniach słów: „piwo”, „jęczmień”, „browar” oraz innych, które przyjdą mu kiedyś do głowy. Chciałby przetestować kilka sposobów wyszukiwania słów. Zaproponujcie odpowiednie rozwiązania”</p>	<p>Wykorzystanie słownika do przechowywania wzorców.</p> <p>Implementacja odpowiednich struktur danych, np. stosów lub tablic,</p>	<p>Algorytm Naiwny [<math>O(n \cdot m)</math>]</p> <p>Algorytm <b>KMP Knutha-Morrisa-Pratta</b> [<math>O(n + m)</math>]</p> <p>Algorytm <b>Rabina-Karpa</b> [średnio <math>O(n + m)</math>]</p> <p>Algorytm <b>Trie</b> [<math>O(n + k)</math>, gdzie <math>k</math> to liczba dopasowań]</p> <p>Algorytm <b>Boyer-Moore</b> [<math>O(n \cdot m)</math>] – gdy dopasowanie nie występuje i znaki często nie pozwalają na duże przesunięcia]</p>
---	--	---	--	---

6	<p><b>Dane:</b></p> <p>Tekst (plik tekstowy) do skompresowania</p> <p><b>Wynik:</b></p> <p>Skompresowany tekst w postaci ciągu bitów (string z '0' i '1')</p> <p>Mapy kodów Huffmana do kodowania i dekodowania</p>	<p>“Komputer ma ograniczone zasoby”</p>	<p>Drzewo binarne Huffmana — węzły przechowujące znak i częstotliwość</p> <p>Kolejka priorytetowa do budowy drzewa Huffmana</p> <p>Mapy znak → kod oraz kod → znak do kompresji i dekompresji</p>	<p>Algorytm <b>Huffmana</b> [<math>O(n \cdot \log n)</math>, <math>n</math> – liczba symboli]</p>
---	---	---	---	---

7	<p><b>Dane:</b></p> <p>Wymagania dotyczące zastosowania wybranego algorytmu</p> <p><b>Wynik:</b></p> <p>Reprezentacja graficzna projektu przy użyciu bibliotek</p>	<p>Wizualizacja projektu z wykorzystaniem obiektów i bibliotek graficznych</p>	<p>Obiekty</p> <p>Biblioteki graficzne</p>	<p>Brak implementacji algorytmu — nacisk na wykorzystanie bibliotek do tworzenia interfejsu i wizualizacji</p>
---	--	--	--	--