

HoBeerSystem

Algorytmy i Struktury danych 2

Wstęp

Projekt niniejszej aplikacji powstał z myślą o mieszkańcach Shire – spokojnej krainy Hobbitów, znanej z gościnności, zamiłowania do piwa i porządku. Wraz z rosnącym zapotrzebowaniem na jęczmień oraz zwiększoną produkcją lokalnych browarów, pojawiła się potrzeba stworzenia narzędzia wspierającego organizację pracy, wyszukiwanie danych oraz zarządzanie informacjami tekstowymi.

Celem projektu było zaprojektowanie i stworzenie aplikacji wspomagającej procesy logistyczne i informacyjne w Shire. System pozwala m.in. na:

- Graficzną prezentację rozmieszczenia pól jęczmienia, browarów i karczm
- Efektywne wyszukiwanie słów kluczowych w dokumentach przy użyciu różnych algorytmów
- Kompresję i Dekompresję danych
- Prostą obsługę interfejsu umożliwiającego Hobbitom intuicyjne korzystanie z narzędzia.

Projekt realizowany był zespołowo, z podziałem na konkretne role (np. implementacja algorytmów, projektowanie interfejsu, dokumentacja). Prace przebiegały zgodnie z harmonogramem uwzględnionym w diagramie Gantta. Kluczowym założeniem było połączenie funkcjonalności technicznych z narracyjnym, fabularnym światem Shire, co pozwoliło uczynić aplikację zarówno użyteczną, jak i osadzoną w spójnym kontekście

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

Specyfikacja wymagań

Poniższa specyfikacja zawiera opis siedmiu zidentyfikowanych problemów rozwiązanych w ramach projektu. Dla każdego problemu przedstawiono kontekst zadania, zastosowane algorytmy i struktury danych, a także cele ich użycia. Specyfikacja służy jako szczegółowy przegląd funkcjonalnych komponentów projektu oraz podstaw do jego implementacji i dalszego rozwoju.

Problem	Specyfikacja problemu (dane i wyniki)	Do jakich treści w zadaniu odnosi się algorytm	Zastosowane struktury danych	Informacje o zastosowanym algorytmie
1	<p>Dane:</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>Wynik:</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>Dane:</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>Wynik:</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 Edmondsa-Karpa [maksymalny przepływ $O(V \cdot E^2)$]</p> <p>BFS (w każdej iteracji przepływu)</p>
---	---	---	---	--

3	<p>Dane:</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<vector<Krawedz>>)</p> <p>Wynik:</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 [$O(F \cdot (V + E \cdot \log V))$] – znajdowanie minimalnego kosztu przepływu</p> <p>Algorytm Dijkstry z potencjałami [$O(E \cdot \log V)$ na iterację] – wyznaczanie najtańszych ścieżek</p>
---	--	---	---	--

4	<p>Dane:</p> <p>Współrzędne punktów granicznych każdej ćwiartki pola (Punkt)</p> <p>Ilość jęczmienia produkowanego na każdej ćwiartce (dane dodatkowe)</p> <p>Wynik:</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 Grahama [wypukła otoczka $O(n \cdot \log n)$]</p>
---	--	---	--	---

5	<p>Dane:</p> <p>Plik tekstowy zawierający treść do przeszukania.</p> <p>Lista poszukiwanych słów (wzorców)</p> <p>Wynik:</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 [$O(n \cdot m)$]</p> <p>Algorytm KMP Knutha-Morrisa-Pratta [$O(n + m)$]</p> <p>Algorytm Rabina-Karpa [średnio $O(n + m)$]</p> <p>Algorytm Trie [$O(n + k)$, gdzie k to liczba dopasowań]</p> <p>Algorytm Boyer-Moore [$O(n \cdot m)$] – gdy dopasowanie nie występuje i znaki często nie pozwalają na duże przesunięcia]</p>
---	--	---	--	---

6	<p>Dane:</p> <p>Tekst (plik tekstowy) do skompresowania</p> <p>Wynik:</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 Huffmana [$O(n \cdot \log n)$, n – liczba symboli]</p>
---	---	---	---	---

7	<p>Dane:</p> <p>Wymagania dotyczące zastosowania wybranego algorytmu</p> <p>Wynik:</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>
---	--	--	--	--

Charakterystyka problemów i zastosowanych algorytmów

Problem 1

Reprezentacja danych o infrastrukturze Shire

W pierwszym problemie kluczowym zadaniem było przygotowanie modelu danych opisujących infrastrukturę Shire, czyli pól jęczmienia, browarów, karczm oraz łączących je dróg. Problem ten stanowił fundament dla dalszych etapów projektu. Celem było stworzenie takiej reprezentacji, która umożliwi późniejsze zastosowanie algorytmów przepływu, optymalizacji czy wizualizacji. Zastosowane struktury danych, takie jak struktury `struct`, kontenery STL `vector` i `map`, pozwoliły na przejrzyste i wydajne przechowywanie informacji o obiektach oraz relacjach między nimi. Dzięki temu możliwe było sprawne zarządzanie danymi wejściowymi i ich dalsze przetwarzanie. Choć w tym etapie nie wykorzystano żadnego algorytmu, to logiczne uporządkowanie danych umożliwiło dalsze rozwijanie aplikacji oraz zapewniło stabilną i elastyczną bazę pod kolejne problemy.

Problem 2

Maksymalizacja przepływu piwa do karczm

Drugi problem dotyczył zwiększenia efektywności dostarczania piwa z browarów do karczm, przy uwzględnieniu istniejących ograniczeń infrastrukturalnych, takich jak przepustowość dróg. Celem było określenie maksymalnej ilości piwa, jaką można przetransportować w systemie, traktowanym jako graf przepływu. W tym celu użyto algorytmu Edmondsa-Karpa, będącego odmianą algorytmu Forda-Fulkersona, opierającą się na BFS-ie do znajdowania ścieżek powiększających. Został wybrany ze względu na swoją przejrzystość, poprawność i względnie dobrą złożoność dla grafów średniej wielkości. Dzięki temu algorytmowi udało się określić maksymalne możliwości przesyłu, co stanowiło punkt odniesienia dla późniejszych prób optymalizacji kosztów transportu.

Problem 3

Minimalizacja kosztu transportu przy maksymalnym przepływie

Po ustaleniu maksymalnych możliwości przesyłu, kolejnym krokiem było uwzględnienie ekonomii – czyli minimalizacja kosztów naprawy i użycia dróg przy zachowaniu maksymalnego przepływu. Cel problemu polegał na zoptymalizowaniu kosztów logistycznych bez strat w efektywności dostaw. Wykorzystano algorytm Successive Shortest Path (SSP), wspierany przez algorytm Dijkstry z zastosowaniem potencjałów. Algorytmy te pozwoliły na efektywne znajdowanie najtańszych ścieżek w grafie o dodatnich wagach. Zastosowanie takiego podejścia wynikało z konieczności rozwiązywania problemu minimum-cost-flow, który klasyczne algorytmy maksymalnego przepływu nie uwzględniają. W wyniku działania algorytmu udało się określić zarówno maksymalny przepływ, jak i przypisany mu minimalny koszt, co dało realistyczną symulację gospodarki logistycznej Shire.

Problem 4

Przetwarzanie danych przestrzennych – analiza ćwiartek pól

Celem tego problemu było przetworzenie danych przestrzennych opisujących poszczególne ćwiartki pól uprawnych, które miały postać rozłącznych wielokątów wypukłych. Dla każdego z nich należało określić jego kształt oraz wielkość, co było kluczowe dla późniejszego uwzględnienia produkcji jęczmienia. W tym celu zastosowano algorytm Grahama, który służy do wyznaczania wypukłej otoczki zbioru punktów. Został on użyty z uwagi na swoją wysoką skuteczność i dokładność w analizie geometrii płaskiej. Dodatkowo użyto wzoru "shoelace formula" do obliczania pól powierzchni wielokątów. Rezultatem było uzyskanie precyzyjnych danych na temat powierzchni ćwiartek i ich ułożenia, co pozwoliło na dokładniejsze modelowanie produkcji jęczmienia i jego dalsze przetwarzanie.

Problem 5

Wyszukiwanie słów w tekście

Ten problem dotyczył automatyzacji i przyspieszenia procesu wyszukiwania konkretnych słów (takich jak „piwo”, „jęczmień”, „browar”) w tekście. Celem było porównanie różnych podejść algorytmicznych do problemu wyszukiwania wzorców. Wykorzystano zarówno algorytmy proste (naiwne), jak i bardziej zaawansowane, takie jak KMP, Rabina-Karpa, Trie oraz Boyera-Moorea. Każdy z nich został użyty, aby zaprezentować różne strategie: KMP zapewnia linearną złożoność względem długości tekstu i wzorca, Trie umożliwia przeszukiwanie wielu wzorców jednocześnie, a Rabina-Karpa testowano ze względu na jego efektywność przy dużych ilościach zapytań. Dzięki temu porównaniu możliwe było zbadanie efektywności wyszukiwania oraz wybór najlepszego algorytmu w zależności od potrzeb. Umożliwiło to hobbitom szybkie i skuteczne przeszukiwanie dokumentów i raportów.

Problem 6

Kompresja danych z użyciem algorytmu Huffmana

W szóstym problemie zajęto się kompresją tekstu z użyciem klasycznego algorytmu Huffmana. Celem było zmniejszenie rozmiaru przechowywanych danych tekstowych przy zachowaniu ich pełnej treści. Ze względu na ograniczone zasoby komputerowe, typowe dla Shire, zależało na maksymalnej efektywności kompresji. Algorytm Huffmana, który opiera się na częstości występowania symboli, został wybrany ze względu na swoją prostotę i optymalność przy kodowaniu bezstratnym. W efekcie możliwe było znaczne zredukowanie objętości plików, co odciążało systemy i umożliwiło łatwiejsze przesyłanie danych między różnymi elementami infrastruktury.

Problem 7

Wizualizacja i interfejs użytkownika

Ostatni problem dotyczył stworzenia graficznego interfejsu użytkownika, który pozwoliłby na prezentację danych oraz wyników działania algorytmów w sposób przejrzysty i estetyczny. Zadanie to miało na celu umożliwienie hobbitom intuicyjne korzystanie z aplikacji, bez konieczności znajomości technicznych aspektów działania programu. W tym celu wykorzystano biblioteki graficzne w środowisku Qt. Choć nie zastosowano tutaj żadnego algorytmu obliczeniowego, ważną rolę odegrało poprawne odwzorowanie danych oraz integracja z wcześniejszymi wynikami. Dzięki wizualizacji możliwe było efektywne zarządzanie infrastrukturą Shire, podejmowanie decyzji oraz prezentacja wyników projektu w sposób zrozumiały dla użytkowników końcowych.

Przebiegu prac

	Tydzien 1	Tydzien 2	Tydzien 3	Tydzien 4	Tydzien 5	Tydzien 6	Tydzien 7	Tydzien 8
Specyfikacja + Testy								
Problem 1								
Problem 2								
Problem 3								
Problem 4								
Problem 5								
Problem 6								
Problem 7								

Opis przebiegu prac

Prace nad projektem rozpoczęły się od wspólnego ustalenia założeń i struktury całego systemu. Etap planowania przebiegł sprawnie – szybko zdefiniowano główne cele projektu, podzielono zadania oraz określono zakres funkcjonalności aplikacji. Na tym etapie nie pojawiły się większe trudności.

Kolejne etapy realizowane były sukcesywnie, zgodnie z harmonogramem. Implementacja pierwszych problemów, takich jak struktury danych, przepływy w grafach czy algorytmy wyszukiwania słów, przebiegała bez większych komplikacji. Zadania zostały podzielone między członków zespołu, co pozwoliło na równoległe prace i szybszy postęp.

Największe trudności pojawiły się podczas implementacji algorytmu wypukłej otoczki (Problem 4). Głównym wyzwaniem okazało się nie tylko jego zaimplementowanie, ale przede wszystkim znalezienie właściwego miejsca w projekcie, w którym jego zastosowanie byłoby uzasadnione i zgodne z wcześniejszą logiką działania systemu.

Zespół musiał dokładnie przeanalizować sposób integracji danych przestrzennych z istniejącą infrastrukturą, co wymagało dodatkowych konsultacji i testów.

Dużym wyzwaniem było również stworzenie graficznego interfejsu użytkownika (Problem 7). Integracja danych z różnych części projektu, prezentacja wyników działania algorytmów oraz zapewnienie intuicyjnej obsługi wymagały dokładnego przemyślenia architektury wizualnej oraz zapoznania się z bibliotekami graficznymi Qt. Było to szczególnie trudne, ponieważ wymagało umiejętności łączenia logiki algorytmicznej z komponentami graficznymi i zarządzaniem stanem aplikacji.

Pomimo napotkanych trudności, projekt zakończył się sukcesem – udało się stworzyć funkcjonalną aplikację, która realizuje wszystkie założone cele i stanowi przydatne narzędzie wspierające pracę hobbitów w Shire.

Wnioski końcowe

Projekt udało się zrealizować zgodnie z początkowymi założeniami. Stworzona aplikacja spełnia wyznaczone cele – usprawnia produkcję i dystrybucję piwa w Shire, umożliwia szybkie przeszukiwanie tekstów oraz oferuje funkcjonalności związane z kompresją danych i wizualizacją wyników. Efekty naszej pracy są zgodne z oczekiwaniami, a każda część systemu została pomyślnie zaimplementowana i zintegrowana.

W trakcie realizacji nauczyliśmy się lepszej współpracy i skuteczniejszego działania zespołowego. Dzięki jasnemu podziałowi ról oraz wzajemnemu wsparciu udało nam się pokonać trudności, które pojawiły się na dalszych etapach projektu.

Choć końcowy efekt jest zadowalający, dostrzegamy też przestrzeń do ulepszeń. W przypadku niektórych problemów można byłoby poprawić wydajność, na przykład poprzez optymalizację złożoności obliczeniowej lub zastosowanie bardziej zaawansowanych struktur danych. W przyszłości warto byłoby również rozbudować interfejs graficzny i dodać nowe funkcje, takie jak:

- edytor mapy Shire z możliwością ręcznego wprowadzania danych,

- symulacje przepływu piwa między browarami a karczmami w czasie rzeczywistym,
- rozbudowany moduł analizy tekstu z możliwością filtrowania i sortowania wyników,
- zarządzanie produkcją i zapasami surowców przez użytkownika.

Projekt ma duży potencjał rozwoju i może być podstawą do stworzenia bardziej kompleksowego systemu wspomagającego zarządzanie surowcami, logistyką i informacją w świecie Shire. Zdobyta wiedza i doświadczenie stanowią solidną bazę do dalszych prac.