

• Tukan Tomek i zaginiona ścieżka •



Wprowadzenie

Tukan Tomek jest znanym podróżnikiem, który uwielbia eksplorować egzotyczne miejsca. Podczas jednej ze swoich wypraw w dżungli, Tomek zgubił swoją ulubioną ścieżkę prowadzącą do magicznej polany, pełnej rzadkich owoców. Twoim zadaniem jest pomóc Tukanowi Tomkowi odnaleźć najkrótszą ścieżkę powrotną na magiczną polanę, korzystając z prostych algorytmów przeszukiwania grafu.

Wymagania

1. Opis problemu:

- Dżungla jest reprezentowana jako graf, gdzie węzły reprezentują miejsca w dżungli, a krawędzie reprezentują ścieżki między nimi.
- Każda krawędź ma przypisaną wagę reprezentującą odległość między miejscami.
- Punktem startowym jest miejsce "S" (start), a celem jest miejsce "P" (polana).

2. Backend:

- a. Stwórz API, które przyjmie dane wejściowe w postaci grafu (węzły, krawędzie i ich wagi). Przykładowy format danych wejściowych znajduje się w pliku “przykładowy graf.json”.
 - b. API powinno zwrócić najkrótszą ścieżkę z punktu "S" do punktu "P" oraz łączną odległość tej ścieżki.
 - c. Użyj prostego algorytmu przeszukiwania grafu, takiego jak algorytm Dijkstry, aby znaleźć najkrótszą ścieżkę.
- 3. Frontend:**
- a. Stwórz prosty interfejs użytkownika, który pozwoli na wprowadzenie grafu (węzłów, krawędzi i ich wag) w sposób graficzny.
 - b. Interfejs powinien wyświetlać znaną ścieżkę na mapie dżungli oraz informować o łącznej odległości ścieżki.
 - c. Dodaj animację, która pokaże, jak Tukan Tomek przemieszcza się po wyznaczonej ścieżce.
- 4. Dodatkowe wymagania:**
- a. Aplikacja powinna umożliwiać zapisywanie i wczytywanie grafów.
 - b. Dodaj funkcję, która pozwoli użytkownikom na dzielenie się swoimi mapami z innymi użytkownikami.
 - c. Zapewnij, że API jest odpowiednio zabezpieczone przed typowymi atakami (np. SQL Injection, XSS).
- 5. Dostarczenie rozwiązania:**
- a. Na wskazany adres email kariera@tukano.com.pl wyślij link do repozytorium GitHub, gdzie zostało umieszczone rozwiązanie.
 - b. Rozwiązanie powinno zawierać plik README.md opisujący jak uruchomić aplikację.
 - c. Rozwiązanie po pobraniu powinno się kompilować i po wykonaniu instrukcji z pliku README.md uruchamiać się bez błędów.

Opis techniczny

- 6. Backend:**
- a. Możesz użyć dowolnej technologii backendowej (np. Node.js, Python (Flask), Java (Spring Boot), .NET).
 - b. Baza danych może być relacyjna (np. MySQL, SQL Server) lub nierelacyjna (np. MongoDB).
- 7. Frontend:**
- a. Możesz użyć dowolnych technologii frontendowych (np. HTML, CSS, JavaScript, React, Angular).
 - b. Upewnij się, że aplikacja jest responsywna i działa na różnych urządzeniach (desktop, tablet, smartfon).

8. **Algorytm przeszukiwania grafu:**

- a. Zaimplementuj algorytm Dijkstry, aby znaleźć najkrótszą ścieżkę w grafie.
[Algorytm Dijkstra, czyli szukanie najkrótszej drogi \(youtube.com\)](#)
[Algorytm Dijkstry \(for dummies\) \(youtube.com\)](#)
- b. Przykładowe dane dla grafu znajdują się w pliku “przykładowy graf.json”.

Pamiętajcie, że nie wszystkie opisane wymagania muszą zostać spełnione, aby wystać nam rozwiązanie. Każda praca będzie oceniona indywidualnie. Na dostarczenie swoich rozwiązań macie czas do poniedziałku **17 czerwca 2024 r., godzina 12:00**.

Życzymy powodzenia w implementacji zadania!