

Heathrow to London

Programowanie funkcyjne - Projekt Haskell

Katarzyna Gajewska
Michał Kreft
Tomasz Biel

Politechnika Krakowska
im. Tadeusza Kościuszki

25 styczeń 2022



Plan prezentacji

1 Zadanie

2 Obliczanie najszybszej ścieżki



Zadanie



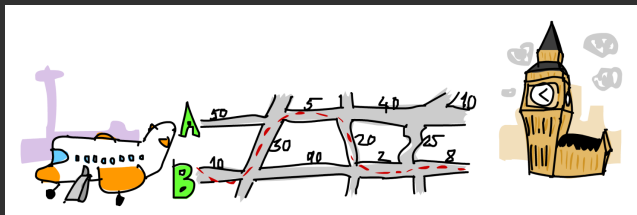
Zadanie

Założmy, że jesteśmy w podróży służbowej. Nasz samolot właśnie wylądował w Anglii i wynajmujemy samochód. Mamy spotkanie bardzo szybko i musimy jak najszybciej dostać się z lotniska Heathrow do Londynu. Z Heathrow do Londynu prowadzą dwie główne drogi i przecina je wiele dróg regionalnych. Podróż z jednego skrzyżowania na drugie zajmuje określoną ilość czasu. Od nas zależy znalezienie optymalnej ścieżki do podjęcia, abyśmy na czas dotarli na spotkanie w Londynie.



Zadanie

Zaczynamy po lewej stronie i możemy albo przejść na drugą główną drogę, albo iść przed siebie.



Rysunek 1: Graficzne przedstawienie problemu

Jak widać na obrazku, najszybszą drogą z Heathrow do Londynu w tym przypadku jest rozpoczęcie na głównej drodze B, przejście przez skrzyżowanie, przejście do przodu na A, ponowne skrzyżowanie, a następnie dwukrotny przejazd do przodu na B. Jeśli weźmiemy tą drogą zajmuje nam 75 minut. Gdybyśmy wybrali inną drogę, zajęłoby to więcej czasu.

Obliczanie najszybszej ścieżki

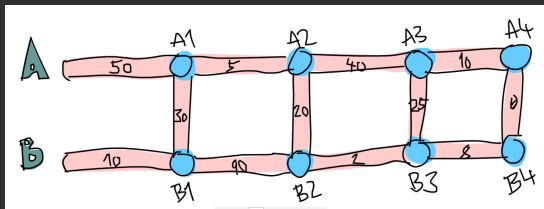


Obliczanie najszybszej ścieżki

Jak więc samodzielnie znaleźć najszybszą drogę z Heathrow do Londynu? Cóż, możemy po prostu spojrzeć na cały obraz i spróbować zgadnąć, jaka jest najszybsza ścieżka i mieć nadzieję, że nasze przypuszczenia są poprawne. To rozwiązanie działa w przypadku bardzo małych danych wejściowych, ale co, jeśli mamy drogę, która ma 10 000 odcinków? Nie będziemy też mogli powiedzieć z całą pewnością, że nasze rozwiązanie jest optymalne; możemy po prostu powiedzieć, że jesteśmy prawie pewni. Więc to nie jest dobre rozwiązanie. Naszym zadaniem jest stworzenie programu, który pobiera dane wejściowe reprezentujące system drogowy i wyświetla najszybszą ścieżkę przez niego.



Obliczanie najszybszej ścieżki



Rysunek 2: Uproszczony rysunek naszego systemu dróg

Jako przykład rozumowania znajdziemy najszybszą drogę do pierwszego skrzyżowania (pierwsza kropka A, oznaczona jako A1). To dość trywialne. Po prostu sprawdzamy, czy szybciej jest przejść bezpośrednio do przodu z punktu A lub z punktu B a następnie w górę. Oczywiście szybciej jest przejść z punktu B, a następnie do góry, ponieważ zajmuje to 40 minut, podczas gdy przejście bezpośrednio przez A zajmuje 50 minut. A co ze skrzyżowaniem B1? Widzimy, że o wiele szybciej jest po prostu przejść bezpośrednio przez B (koszt 10 minut), ponieważ przejście przez A, a następnie przejście przez nią zajęłoby nam 80 minut. Można rozpisywać się w podobny sposób na temat kolejnych skrzyżowań lecz to już będzie zadanie naszego programu, który wykorzysta właśnie w ten sposób działającą metodę sprawdzania najkrótszej drogi.



Obliczanie najszybszej ścieżki

Podsumowując, aby uzyskać najlepszą drogę z Heathrow do Londynu, robimy następujące kroki:

1. Sprawdzamy, jaka jest najkrótsza droga do następnego skrzyżowania na głównej drodze A. Dwie opcje to pójście prosto do przodu lub rozpoczęcie od przeciwnej drogi, pójście do przodu, a następnie przejście w bok. Zapamiętujemy jaka to ścieżka i ile zajmuje.
2. Używamy tej samej metody, aby znaleźć najlepszą ścieżkę do następnego skrzyżowania na głównej drodze.
3. Sprawdzamy, czy droga do następnego skrzyżowania zajmuje mniej czasu, jeśli jedziemy z poprzedniego skrzyżowania, czy też idziemy z poprzedniego skrzyżowania, a następnie przechodzimy na drugą stronę. To samo robimy na przeciwległym skrzyżowaniu.
4. Robimy to dla każdej sekcji, aż dojdziemy do końca.
5. Gdy dotrzemy do końca, szybsza z dwóch dostępnych ścieżek jest naszą optymalną ścieżką.

Tak więc w istocie otrzymujemy jedną najszybszą ścieżkę na drodze A i jedną najszybszą ścieżkę na drodze B. Kiedy docieramy już do końca, jedna z tych dwóch ścieżek jest właśnie tą najszybszą.



Dziękujemy za uwagę

Przejdźmy do implementacji

