

Sztuczna Inteligencja

Wizualizacja algorytmów przeszukiwania

Jakub Jabłoński
184398

Krystian Jandy
184589

23 maja 2022

Wprowadzenie

Metody szukania są bardzo przydatnym narzędziem w informatyce. Mają one szerego zastosowań począwszy od szukania najkrótszej ścieżki (np. pomiędzy dwoma miastami), aż po rozbudowane algorytmy systemów sztucznej inteligencji używane m.in. w grach strategicznych oraz w systemach inteligentnych domów, pojazdów i w wielu innych aspektach informatycznych.

W algorytmach szukania zwracamy uwagę na różne czynniki, w zależności od potrzeby. Wśród dostępnych algorytmów wyróżniamy takie, które w swoim działaniu uwzględniają m. in. obciążenie krawędzi lub węzłów (Dijkstra, A*), bądź skupiają się na jak najszybszym przeszukiwaniu grafu i minimalizacji liczby iteracji (BFS, DFS).

BFS - Metoda przeszukiwania wszerz

BFS jest jedną z metod przeszukiwania grafów. Jego działanie charakteryzuje się tym, że szukanie zaczyna się od konkretnego węzła, zaś następnym krokiem jest odwiedzenie wszystkich sąsiadów tego wierzchołka. Czynność ta jest powtarzana, aż do momentu przejścia przez wszystkie wierzchołki w grafie.

Złożoność

Złożoność algorytmu BFS: $O(n + m)$, gdzie:

- n - liczba wierzchołków
- m - liczba krawędzi

Zastosowanie

- Sprawdzenie dwudzielności grafu
- Odnalezienie wszystkich połączeń w grafie
- Komunikacja peer to peer - do znalezienia sąsiadów
- W wyszukiwarkach stron internetowych
- GPS - znajdowanie pobliskich obiektów lub które osób, które możemy znać
- Garbage Collector - z użyciem algorytmu Cheney'a używany do kopiowania referencji.

DFS (Metoda przeszukiwania w głąb)

Podobnie jak BFS służy do przeszukiwania grafów, natomiast polega on na przeszukiwaniu grafu w głąb. Używany w momencie, gdy wiemy, że potrzeba większej liczby iteracji do znalezienia rozwiązania lub gdy nasz graf jest szeroki. Działanie metody DFS charakteryzuje się tym, że przechodzimy do kolejnych sąsiadów wierzchołków począwszy od wierzchołka startowego, aż do momentu, gdy wszystkie zostaną odwiedzone.

Złożoność

Złożoność algorytmu dfs: $O(n + m)$

Zastosowanie

- Wykrywanie spójnych składowych w grafie.
- Umożliwia sprawdzenie czy pomiędzy węzłami istnieje ścieżka.
- Pozwala zbadać spójność grafu.
- Używany w topologicznym sortowaniu - do przydzielania rozkładu pracy
- Do rozwiązywania zagadnień, gdzie istnieje tylko jedno rozwiązanie (labirynt)

Dijkstra

Metoda przeszukiwania Dijkstry umożliwia znalezienie najkrótszej ścieżki pomiędzy dwoma węzłami w grafie. Wymaganiem algorytmu jest to, aby krawędzie nie były obciążone ujemnymi wagami. Algorytm Dijkstry jest algorytmem zachłannym, jego działanie polega na tym, że dla sąsiadów aktualnego wierzchołka sprawdzamy, czy koszt dojścia z aktualnego wierzchołka do sąsiada zsumowany z kosztem dojścia do aktualnego węzła jest niższy niż aktualnie wpisany, jak warunek jest spełniony to aktualizujemy tę wartość i informację skąd przyszedliśmy. W momencie wykonania tego dla wszystkich sąsiadów, wybieramy węzeł o najniższym koszcie, który nie był jeszcze wybrany. Algorytm jest wykonywany do momentu sprawdzenia wszystkich węzłów. Wynikiem działania algorytmu są wszystkie najkrótsze ścieżki pomiędzy węzłami w przetwarzanym grafie wraz z kosztem przejścia przez te ścieżki.

Złożoność

Złożoność algorytmu: $O(n^2)$, a w przypadku zastosowania kolejki priorytetowej: $O(n \log n)$

Zastosowanie

- Używany do wyznaczania trasy do wysłania pakietów danych w sieci komputerowych
- W nawigacji, np. google maps do znalezienia najkrótszej drogi pomiędzy obiektami
- W sieciach telefonicznych (szukanie przejścia transmisji połączenia pomiędzy odbiornikami)
- Wykorzystywany w robotach do wyznaczenia drogi
- Social Media - używany do pokazywania proponowanych znajomych
- Do minimalizacji w przesyle plików w sieciach LAN

A*

Metoda szukania, której wynikiem jest najkrótsza ścieżka pomiędzy dwoma wierzchołkami (używana w grafie ważonym). W odróżnieniu od algorytmu Dijkstry wykorzystuje ona heurystykę (dodatkową informację estymującą najkrótszą drogę od węzła przetwarzanego do końcowego), pozwala to na zredukowanie liczby generowanych węzłów w grafie.

Zastosowanie

- Pozwala na wyznaczenie ścieżki pomiędzy dwoma węzłami, dzięki wprowadzeniu dodatkowej heurystyki jest bardziej wydajny od algorytmu Dijkstry
- Często używany w grach do generowania ścieżek pomiędzy węzłami
- Do generowania trasy dla robotów