Projektowanie Algorytmów i Metody Sztucznej Inteligencji

Temat: Grafy i algorytm Dijkstry

Nazwisko i Imię prowadzącego kurs: mgr inż. Marta Emirsajłow

Wykonawca:	
Imię i Nazwisko Nr indeksu, wydział	Jakub Kolasa 249012, W4
Termin zajęć	Pt, 13:15-15:00
Data oddania sprawdozdania	10.05.2020

1. Wprowadzenie.

Celem projektu jest przeprowadzenie analizy algorytmu najkrótszej ścieżki w grafie. Badanym będzie algorytm Dijkstry. Badanie efektywności algorytmu (czasu działania) będzie odbywać się dla różnych liczb wierzchołków oraz gęstości grafu. Przeanalizowany zostanie również wpływ implementacji grafu: z użyciem listy sąsiedztwa lub macierzy sąsiedztwa.

Algorytm Dijkstry stosuje się w grafach nieskierowanych, z wagami nieujemnymi. Znajduje on najkrótszą drogę między wierzchołkami grafu. Pesymistyczna złożoność obliczeniowa algorytmu wynosi **O(E + VlogV)**, gdzie V to liczba wierzchołków, a E - liczba krawędzi. Algorytm działa najefektywniej, gdy wykorzystuje się w nim kolejkę priorytetową.

2. Krótki opis wykorzystanego programu.

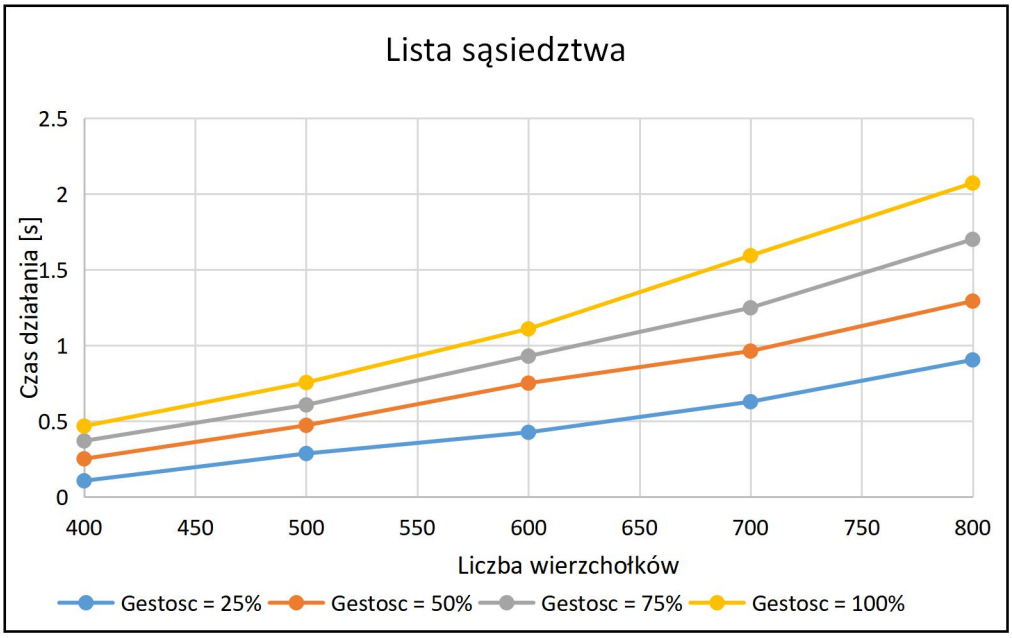
W celu wykonania badań zaimplementowano oba rodzaje wcześniej opisanych grafów. Są to grafy ważone, nieskierowane. Klasy te mają możliwość zapisu/wczytywania danych z pliku. Algorytm Dijkstry w celu lepszej efektywności korzysta z kolejki priorytetowej opartej na strukturze kopca.

3. Badania algorytmu Dijkstry.

Badania przeprowadzano dla 50 powtórzeń wykonania algorytmu dla różnej liczby wierzchołków oraz gęstości grafu.

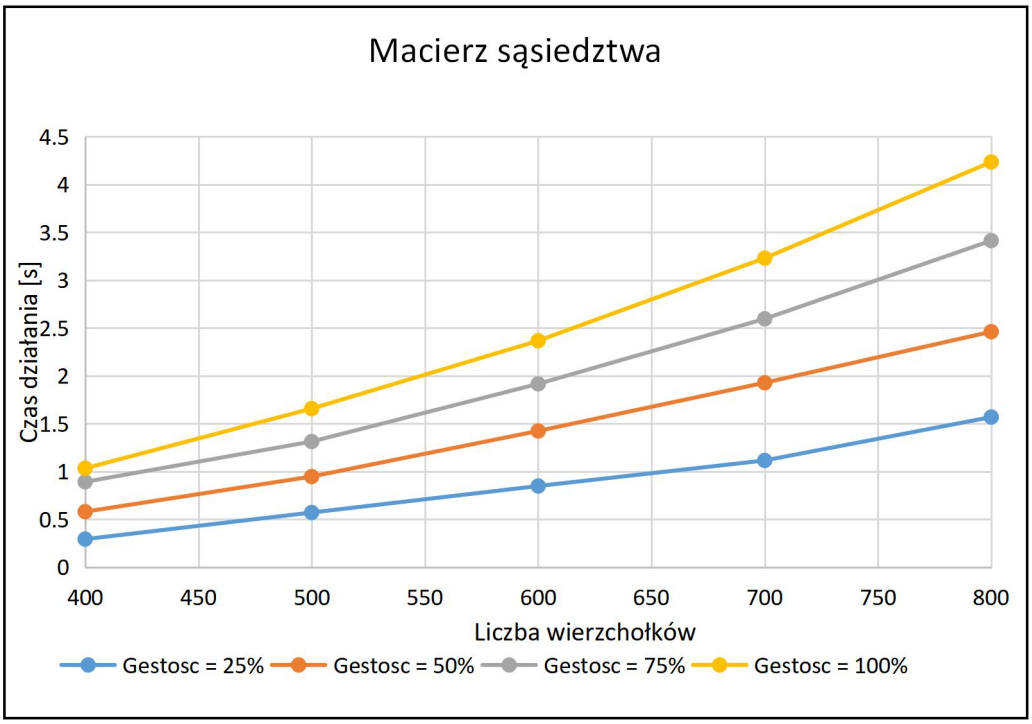
Wyniki otrymane w wyniku badaia czasu trwana algorytmu dla listy sąsiedztwa

Lista sąsiedztwa	Gęstość = 25%	50%	75%	100%
V=400	0,107416 s	0,252566 s	0,370801 s	0,468232 s
500	0,287367 s	0,473398 s	0,608151 s	0,755951 s
600	0,427277 s	0,751794 s	0,930158 s	1,1099 s
700	0,629161 s	0,963325 s	1,24919 s	1,59343 s
800	0,905306 s	1,29336 s	1,70088 s	2,07237 s



Wyniki otrymane w wyniku badaia czasu trwana algorytmu dla macierzy sąsiedztwa

Macierz sąsiedztwa	Gęstość = 25%	50%	75%	100%
V=400	0,292526 s	0,579468 s	0,891829 s	1,03289 s
500	0,570636 s	0,946627 s	1,31241s	1,65673 s
600	0,847884 s	1,42316 s	1,91556 s	2,36595 s
700	1,11495 s	1,92747 s	2,59524 s	3,22929 s
800	1,56826 s	2,46046 s	3,41241 s	4,23421 s



4. Analiza oraz wnioski końcowe.

Na podstawie wyników testów można stwierdzić, że algorytm Dijkstry działa lepiej w przypadku zastosowania listy sąsiedztwa przy implementacji grafu. Prawdopodobna różnica czasowa wynika z częstego wykorzystywania metody incidentEdges w algorytmie. Lista sąsiedztwa zwraca wcześniej już ,,przygotowaną" liste incydentnych krawędzi. Natomiast macierz sąsiedztwa musi za każdym razem taką liste tworzyć. W celu skrócenia czasu wykonywania algorytmu w klasie graf_ms zawarto dynamiczą listę, która zawiera w sobie efekt ostatniego użycia metody incidentEdges. Sprawiło to gwałtowny spadek czasu wykonywania się programu.

Graf zbudowany przy pomocy listy sąsiedztwa zajmuje więcej pamięci jednak dzięki temu znacznie przyśpiesza działanie algorytmu Dijkstry.