

PROJEKTOWANIE EFEKTYWNYCH ALGORYTMÓW

PROJEKT

07/10/2021

252736 Hutnik Szymon

Brute Force (1)

Strona	Spis treści
	Treść zadania
	Opis metody
	Opis algorytmu
	Dane testowe
	Procedura badawcza
	Wyniki
	Analiza

1. Treść zadania

Opracować, napisać, zbadać rozwiązanie problemu komiwożacza w wersji optymalizacyjnej algorytmem programowania dynamicznego.

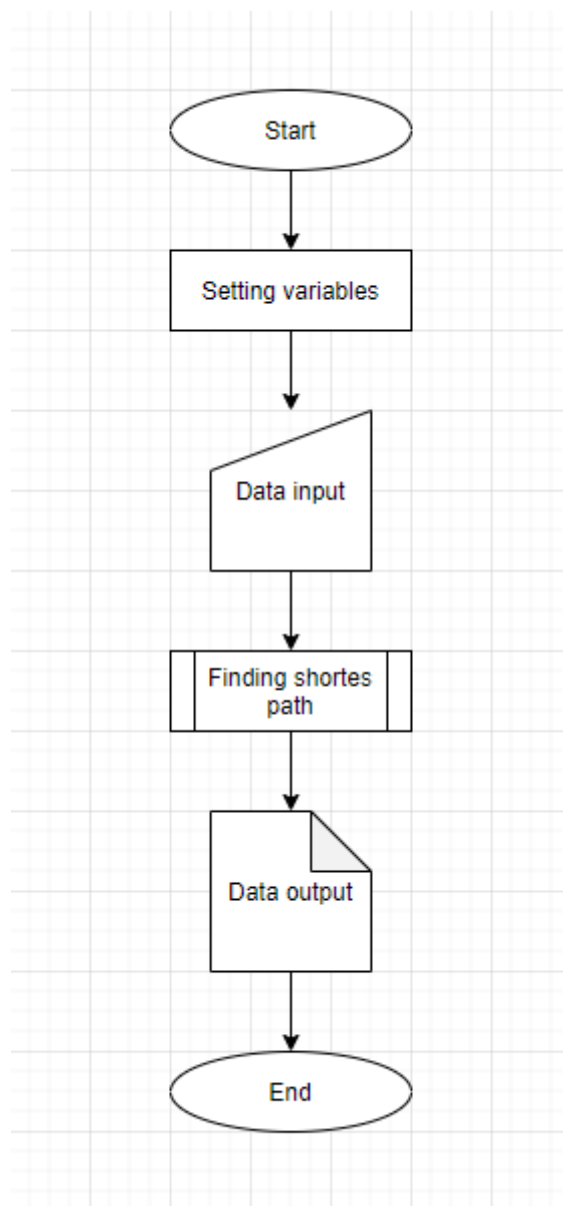
Problem komiwożacza (Travelling Salesman problem) polega na znalezieniu minimalnego cyklu Hamiltona (przejście przez wszystkie wierzchołki tylko raz, startując i kończąc w tym samym punkcie) w pełnym grafie ważonym.

2. Opis metody

Metoda programowania dynamicznego (eng. dynamic programming) polega na podzieleniu problemu na mniejsze części, a następnie rozwiązanie każdej z nich. Każdy następny etap wykorzystuje otrzymane wcześniej wyniki co pozwala skrócić czas obliczenia całości, bo nie ma potrzeby wykonywania kilkakrotnie tych samych obliczeń. Metoda jest trudniejsza do zaimplementowania i zazwyczaj limitem jest pamięć, a nie czas. W tym problemie sprowadza się ona do znalezienia wszystkich możliwych podścieżek, a następnie „rozbudowywaniu” ich o kolejne węzły, aż otrzymamy pełny cykl i wybierzemy najtańszy z nich.

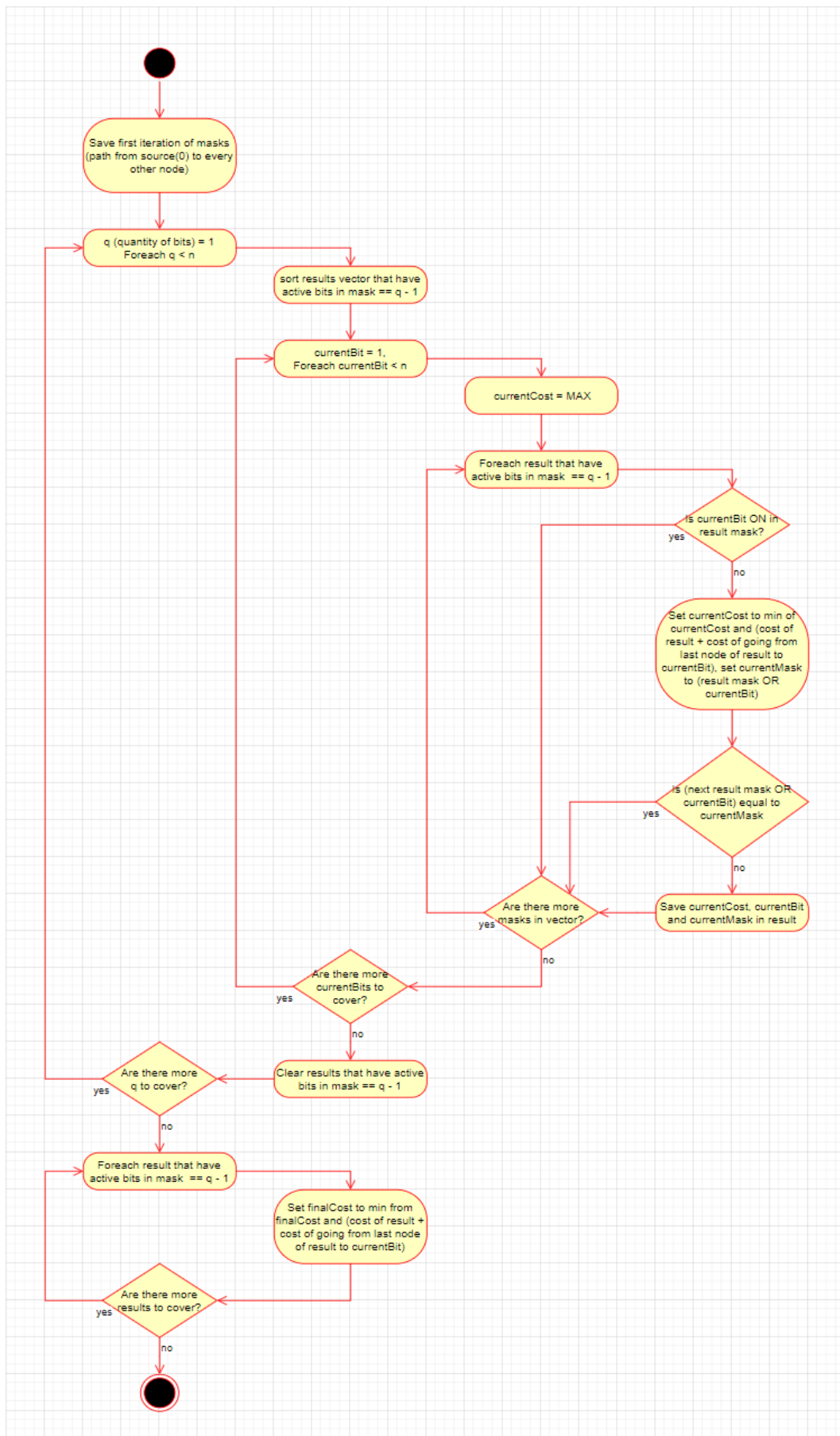
3. Opis algorytmu

Rozwiązanie zaimplementowano w postaci programu opisanego przez poniższy diagram:



Najpierw inicjalizowane są zmienne, najlepsza ścieżka jest ustawiona na maksymalną wartość (INT_MAX), maksymalna maska jest zerowana (otrzyma wartość 2^n-1). Po wczytaniu danych z konsoli następuje uruchomienie właściwej części algorytmu, następnie wypisywany jest wynik oraz czas wykonania właściwego algorytmu.

Właściwą część algorytmu opisuje diagram:



4. Dane testowe

Dane, na których była badana efektywność algorytmu pochodzą ze zbioru udostępnionego przez dr Rudego. Do badania użyto wartości z następujących plików:

- gr17.tsp
- gr21.tsp
- ulysses22.stp
- gr24.stp
- fri26.stp
- bayg29.stp

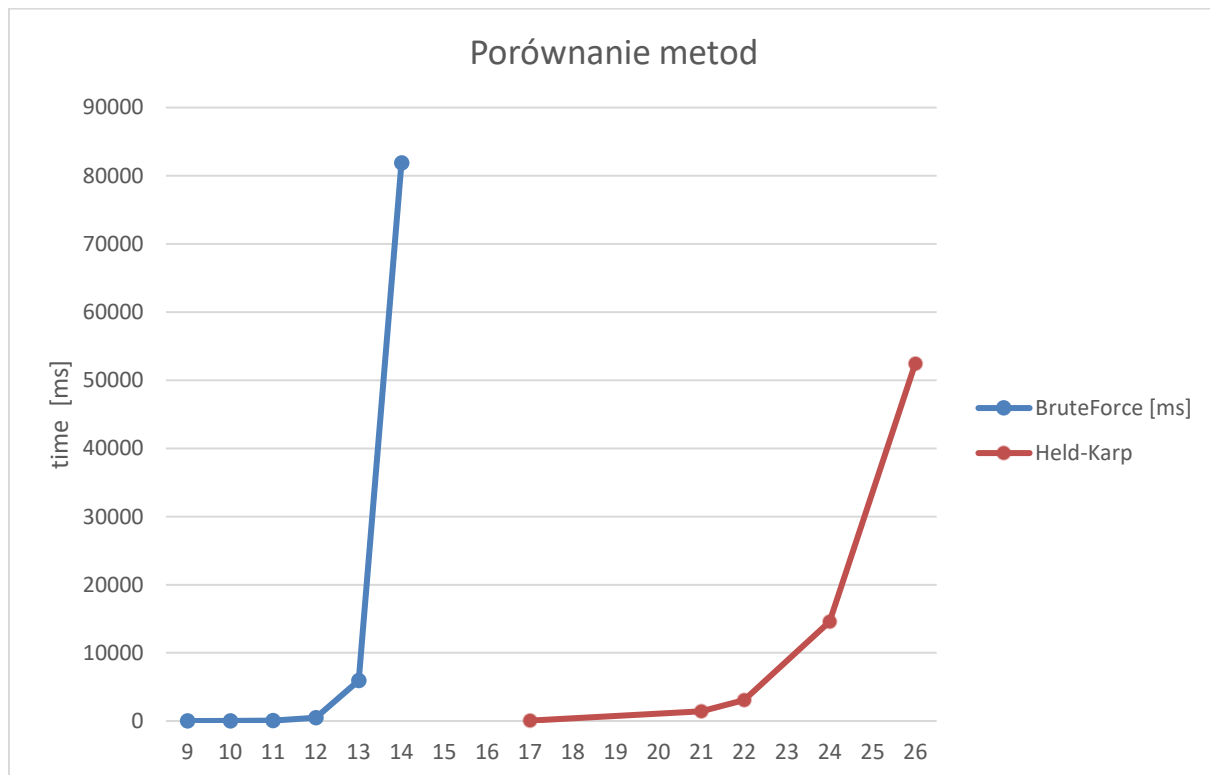
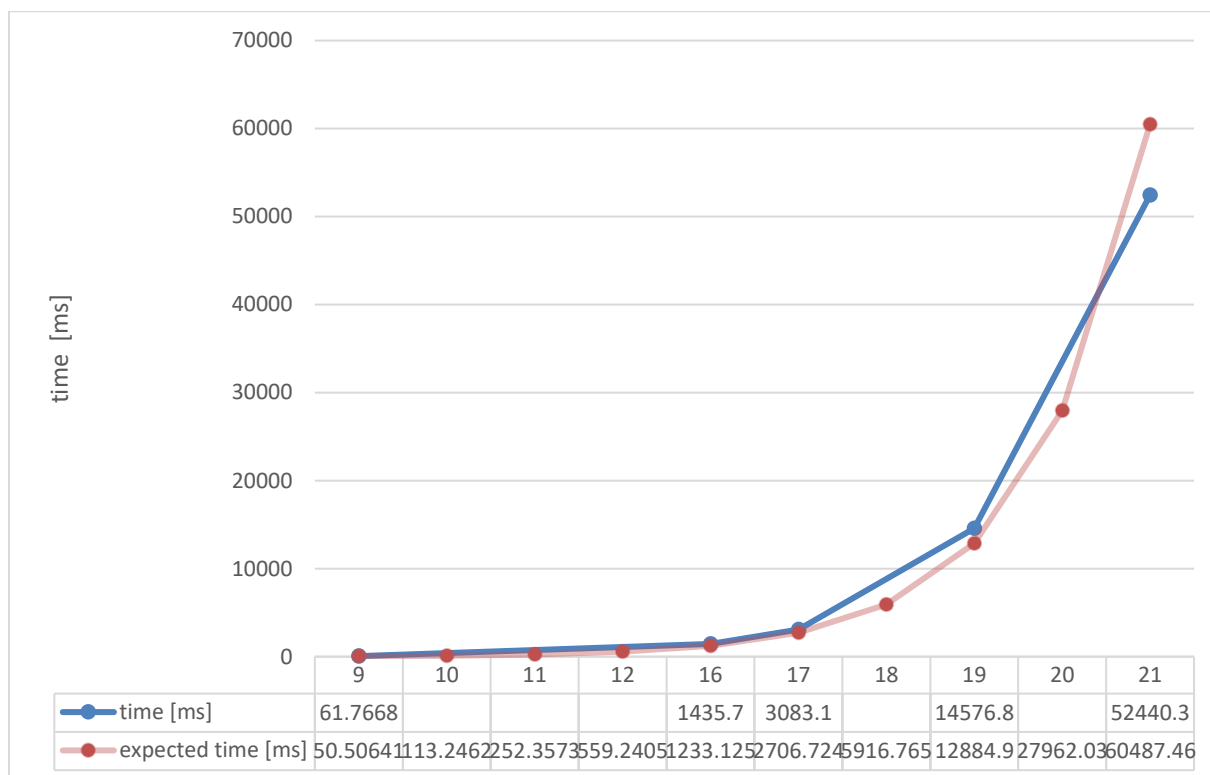
5. Procedura badawcza

Należało zbadać zależność czasu rozwiązania problemu od wielkości instancji. W przypadku algorytmu realizującego przegląd zupełny przestrzeni rozwiązań dopuszczalnych nie występowały parametry programu, które mogły mieć wpływ na czas i jakość uzyskanego wyniku. W związku z tym procedura badawcza polegała na uruchomieniu programu i wklejeniu do niego danych z plików wybranych do badania.

Każda z instancji została wykonana do 10 razy, aby uśrednić czasy, limitem okazało się $n=29$, mimo usuwania poprzednich wyników zajęta pamięć (według Visual Studio 2019) przekroczyła 22GB, co poskutkowało memory overflow i niepowodzeniem testu. Wyniki były zapisywane w Excelu, następnie na ich podstawie została przeprowadzona analiza.

6. Wyniki

Graf zrealizowano w programie Excel:



*dane testowe nie są kolejnymi wartościami n stąd połączenie znanych wyników linią

7. Analiza

Krzywa wzrostu czasu względem wielkości instancji ma charakter wykładniczy (rysunek 1). Nałożenie krzywej expected time ($f(n) = 2^n * n^2$) potwierdza, że badany algorytm wyznacza rozwiązania problemu komiwojażera dla badanych instancji w czasie $2^n * n^2$ zależnym od wielkości instancji (obie krzywe są zgodne co do kształtu). Złożoność czasowa opracowanego algorytmu wynosi $O(2^n * n^2)$. Warto zwrócić uwagę, że nawet wykorzystując dynamiczne struktury danych złożoność pamięciowa skutecznie ogranicza użyteczność tego algorytmu.

Porównanie dotychczas zrealizowanych algorytmów pokazuje jak dużą przewagę ma podejście dynamiczne nad siłowym. Czas wykonania Helda-Karpa zauważalnie niższy jednak jest to okupione wysokim zużyciem pamięci.