

---

## Sprawozdanie z testowania algorytmu FORDA-FULKERSONA

Testowanie i weryfikacja oprogramowania 2012/2013 — Projekt

---

<b>Data</b>	2012-12-25
<b>Wersja</b>	0.2.2
<b>Autorzy</b>	TC, MM, MO, RW

---

## 1 Plan testowania

### 1.1 Wprowadzenie

Testowaniu podlega system realizujący algorytm FORDA-FULKERSONA służący do wyznaczania maksymalnego przepływu w sieciach. Testowane elementy pochodzą z kodu załączonego wraz z książką *Algorytmy. Almanach*. Celem testowania jest zweryfikowanie poprawności już zaimplementowanej funkcjonalności.

### 1.2 Testowane elementy

Testowaniu podlega projekt `ford-fulkerson-test-suite` zawierający implementację algorytmu FORDA-FULKERSONA. Testowana wersja projektu znajduje się w repozytorium, w rewizji `e5ebef7` o nazwie `testing-final`.

### 1.3 Testowana funkcjonalność

Uwaga procesu testowania będzie skupiona na konkretnych implementacjach systemu:

- dla klasy `FlowNetwork` — implementacja `FlowNetworkArray`,
- dla klasy `Search` — implementacja `DFS_SearchArray`.

W chwili obecnej dostępne są też alternatywne implementacje operujące na listach, odpowiednio `FlowNetworkAdjacencyList` oraz `DFS_SearchList`. Wyszczególnienie implementacji jest istotne dla planowania testów biało-skrzynkowych. Testy czarno-skrzynkowe nie mają wiedzy o wybranej implementacji, powinny działać niezależnie od wyboru.

### 1.4 Testowana funkcjonalność — wyłączenia

Do zakresu testów nie wchodziły moduły pomocnicze:

- pakiet `algs.heap`,
- pakiet `algs.list`.

Oraz moduły nie wykorzystywane przez testowany system algorytmu:

- moduł `algs.network.generator`.

### 1.5 Podejście

Środowisko testowe zakłada korzystanie z:

- NetBeans IDE 7.2.1
- Oracle JDK 1.7\_u9
- TestNG 6.8

Specyficzne metodyki, narzędzia i kryteria satysfakcji są opisane w sekcji 2.

## 2 Szczegółowy plan testowania

### 2.1 Testy czarno-skrzynkowe

#### 2.1.1 Wprowadzenie

Cel testów czarno-skrzynkowych nie różni się od wcześniejszych założeń. Testy czarno-skrzynkowe koncentrują się na weryfikacji wyników działania algorytmu FORDA-FULKERSONA dla określonych danych wejściowych.

#### 2.1.2 Testowane elementy

Głównym obiektem testów czarno-skrzynkowych jest klasa `FordFulkerson` z projektu `ford-fulkerson-test-suite`, zawierająca implementację algorytmu FORDA-FULKERSONA. W trakcie testów bezpośrednio wykorzystywane są również inne klasy wchodzące do projektu `ford-fulkerson-test-suite`.

#### 2.1.3 Testowana funkcjonalność

Zakres testów czarno-skrzynkowych obejmuje:

1. Działanie konstruktora klasy `FordFulkerson`.
2. Możliwość wykonania metody `compute()` klasy `FordFulkerson`.
3. Poprawność wyników działania implementacji algorytmu FORDA-FULKERSONA w klasie `FordFulkerson`. Weryfikacja polega na porównaniu wartości zwracanych przez `toString()` po wykonaniu metody `compute()`, do wartości zwracanych przez wyrocznię.

#### 2.1.4 Testowana funkcjonalność — wyłączenia

Z testowania czarno-skrzynkowego wyłączone zostaną funkcjonalności:

- Z klas niewykorzystywanych ze względu na wybór implementacji:
  - `DFS_SearchList`,
  - `FlowNetworkAdjacencyList`,
  - `ShortestPathArray`.
- Z klas abstrakcyjnych, po których dziedziczą wykorzystywane klasy:
  - `FlowNetwork`
  - `Search`
- Z klas reprezentujących szczegóły implementacji:
  - `EdgeInfo`,
  - `DFS_SearchArray` – implementująca algorytm przeszukiwania DEPTH-FIRST SEARCH na implementacji sieci przepływu z klasy `FlowNetworkArray`, której instancja jest wymagana jako drugi argument konstruktora klasy `FordFulkerson`;
  - `VertexInfo`,
  - `VertexStructure`.

#### 2.1.5 Podejście

Podstawą do projektu testów czarno-skrzynkowych jest wyróżnienie klas abstrakcji. Odpowiedni dobór klas pozwoli przetestować implementację algorytmu FORDA-FULKERSONA w rozsądnym zakresie, jednocześnie unikając testowania totalnego. Zidentyfikowane zostały następujące klasy abstrakcji:

1. Ze względu na ilość węzłów:
  - (a) Ujemna ilość węzłów.
  - (b) Brak jakichkolwiek węzłów.

- (c) Dokładnie jeden węzeł, źródłowy, pośredni lub ujście.
  - (d) Sieć przepływowa bez węzłów pośrednich.
  - (e) Sieć przepływowa z jednym węzłem pośrednim – jako minimalna ilość węzłów pośrednich w sieci przepływowej z węzłami pośrednimi.
  - (f) Sieć przepływowa z kilkoma węzłami pośrednimi – jako nieograniczona ilość węzłów w sieci przepływowej z węzłami pośrednimi.
  - (g) Sieć przepływowa z  $1e9$  węzłami pośrednimi – jako maksymalna ilość węzłów pośrednich w sieci przepływowej z węzłami pośrednimi.
  - (h) Ilość węzłów mniejsza od zadeklarowanej.
  - (i) Ilość węzłów większa od zadeklarowanej.
2. Ze względu na pętle:
- (a) W źródle.
  - (b) W węźle pośrednim.
  - (c) W ujściu.
  - (d) Bez pętli.
3. Ze względu na krawędzie bezpośrednie pomiędzy źródłem, a ujściem:
- (a) Ze źródła do ujścia.
  - (b) Z ujścia do źródła.
  - (c) Bez krawędzi pomiędzy źródłem, a ujściem.
4. Ze względu na przepustowość krawędzi:
- (a) Dodatnia.
  - (b) Zerowa.
  - (c) Ujemna.
5. Ze względu na poprawność krawędzi:
- (a) Poprawne krawędzie.
  - (b) Krawędzie zaczynające się w nieistniejącym węźle.
  - (c) Krawędzie kończące się w nieistniejącym węźle.
6. Ze względu na istnienie krawędzi wielokrotnionych:
- (a) Bez krawędzi wielokrotnionych.
  - (b) Z bezpośrednią wielokrotnioną krawędzią pomiędzy źródłem, a ujściem.
  - (c) Z krawędziami wielokrotnionymi o przeciwnych zwrotach.

## **2.2 Testowanie biało-skrzynkowe**

### **2.2.1 Wprowadzenie**

Kryterium satysfakcji przeprowadzanych testów wyznaczane jest na podstawie współczynnika pokrycia kodu. Minimalną dopuszczalną wartością jest 90%. Część testów została dostarczona razem z systemem.

### **2.2.2 Testowane elementy**

Lista poszczególnych klas poddawanych testowaniu oraz pokrycie kodu zapewnione przez testy zawarte wraz z kodem źródłowym.

Pakiet algs.network	
Klasa	Aktualne pokrycie
FordFulkerson	100%
FlowNetwork	100%
FlowNetworkAdjacencyList	75%
FlowNetworkArray	94%
Search	100%
DFS_SearchList	100%
DFS_SearchArray	100%
EdgeInfo	100%
VertexInfo	100%
VertexStructure	93%

Załącznik: Raport obecnego stanu pokrycia.

### 2.2.3 Funkcje wyłączone z testowania

Lista nieużywanych metod dostarczanych przez testowane klasy:

- FlowNetworkAdjacencyList.getCost()
- Example.main()

### 2.2.4 Podejście

Testowanie zostanie przeprowadzone techniką testowania strukturalnego. Głównym kryterium zaliczenia testów będzie współczynnik pokrycia linii kodu, którego wartość musi przekraczać 90%.

Dodatkowo do wytycznych środowiska testowego ustalonych globalnie użyte zostaną:

- Cobertura plugin 1.9.4.1

### 2.2.5 Zadania testowania

Zadanie	Osoba odpowiedzialna
Projekt testów	MO
Przygotowanie przypadków testowych	MO
Implementacja testów	MO
Uruchomienie testów i weryfikacja wyników	MO, TC
Akceptacja wyników przebiegu testowania	TC

## 3 Projekt testów

### 3.1 Testy czarno-skrzynkowe

Poniżej opisane są przypadki testowe wybrane do realizacji w ramach testów czarno-skrzynkowych.

### 3.1.1 Sieć bez węzłów

Sieć bez żadnych węzłów nie istnieje, nie da się dla niej wyznaczyć maksymalnego przepływu.

### 3.1.2 Sieć z 1 węzłem

Za szczególny przypadek sieci przepływu można uznać sieć składającą się z dokładnie jednego węzła. Niezależnie od pozostałych parametrów, taka sieć powinna zostać odrzucona jako nieprawidłowa. Wyróżnić można warianty złożone z:

- Samego źródła.
- Węzła pośredniego.
- Ujścia.
- Wspólnego źródła i ujścia.
- Wspólnego źródła i ujścia z pętlą.

Szczególną uwagę należy zwrócić na ostatni z wymienionych wariantów, który jako jedyny posiadający krawędź, którą można traktować jako krawędź ze źródła do ujścia.

Można wyróżnić cztery schematy sieci z pojedynczym węzłem, każdy w czterech wariantach: bez pętli, z pętlą o dodatniej/ujemnej/zerowej przepustowości, co przekładałoby się na szesnaście przypadków testowych. W ramach planu ograniczono się do pięciu najistotniejszych kombinacji.

### 3.1.3 Sieć bez węzłów pośrednich

Sieć złożona ze źródła i ujścia, bez jakichkolwiek węzłów pośrednich. Wyróżnione zostały następujące przypadki testowe dla topologii sieci z węzłami:

- Nie połączonymi żadną krawędzią.
- Połączonymi pojedynczą krawędzią skierowaną od źródła do ujścia o dodatniej przepustowości.
- Połączonymi pojedynczą krawędzią skierowaną od źródła do ujścia o dodatniej przepustowości z pętlą o dodatniej przepustowości w ujściu.
- Połączonymi pojedynczą krawędzią skierowaną od źródła do ujścia o dodatniej przepustowości z pętlą o ujemnej przepustowości w źródle.
- Połączonymi pojedynczą krawędzią skierowaną od źródła do ujścia o zerowej przepustowości.
- Połączonymi pojedynczą krawędzią skierowaną od źródła do ujścia o ujemnej przepustowości.
- Połączonymi pojedynczą krawędzią skierowaną od ujścia do źródła o dodatniej przepustowości.
- Połączonymi pojedynczą krawędzią skierowaną od ujścia do źródła o ujemnej przepustowości.
- Z wieloma krawędziami skierowanymi od źródła do ujścia.
- Z wieloma krawędziami skierowanymi od ujścia do źródła.
- Z wieloma krawędziami skierowanymi w różnych stronach.

Można bez problemu zdefiniować ponad sto wariantów topologii sieci przepływowej bez węzłów pośrednich. Trzy ze względu na rodzaj przepustowości krawędzi (dodatnia/ujemna/zerowa), siedem ze względu na istnienie i rodzaj pętli (brak/w źródle z dodatnią przepustowością/w źródle z ujemną przepustowością/w źródle z zerową przepustowością/w ujściu z dodatnią przepustowością/w ujściu z ujemną przepustowością/w ujściu z zerową przepustowością), oraz sześć wariantów połączeń pomiędzy krawędziami (bez krawędzi, pojedyncza od źródła do ujścia, pojedyncza od ujścia do źródła, zwielokrotniona ze źródła do ujścia, zwielokrotniona z ujścia do źródła, z pomieszanymi zwrotami). Do przetestowania zostało wybranych 11 najbardziej reprezentatywnych topologii sieci bez węzłów pośrednich.

### 3.1.4 Sieć z 1 węzłem pośrednim

Prosta sieć przepływu zbudowana ze źródła, jednego węzła pośredniego i ujścia. Wyróżnione zostały następujące przypadki testowe w zależności od topologii:

- Połączone pojedynczymi krawędziami skierowanymi ze źródła do węzła pośredniego i z węzła pośredniego do ujścia.
- Z pojedynczymi krawędziami skierowanymi z ujścia do węzła pośredniego i z węzła pośredniego do źródła.
- Z pojedynczymi krawędziami skierowanymi z węzła pośredniego do ujścia i z węzła pośredniego do źródła.
- Z pojedynczymi krawędziami skierowanymi ze źródła do węzła pośredniego i z ujścia do węzła pośredniego.
- Połączone pojedynczymi krawędziami skierowanymi ze źródła do węzła pośredniego i z węzła pośredniego do źródła z dodatkową krawędzią o dodatniej przepustowości ze źródła do ujścia.
- Połączone pojedynczymi krawędziami skierowanymi ze źródła do węzła pośredniego i z węzła pośredniego do źródła z pętlą o dodatniej przepustowości w węźle pośrednim.
- Połączone zwielokrotnionymi krawędziami ze źródła do węzła pośredniego i z węzła pośredniego do źródła, z mieszanymi zwrotami.
- Połączone pojedynczą krawędzią skierowaną ze źródła do węzła pośredniego, bez krawędzi do ujścia.
- Połączone pojedynczą krawędzią skierowaną ze źródła do węzła pośredniego, oraz krawędzią ze źródła do ujścia.
- Połączone pojedynczą krawędzią skierowaną z węzła pośredniego do ujścia, bez połączenia ze źródłem.

W przypadku sieci z jednym węzłem pośrednim można wyróżnić nawet ponad tysiąc różnych rodzajów topologii sieci. Dla dwóch krawędzi istnieje już dziewięć kombinacji rodzajów przepustowości (dodatnia/ujemna/zerowa), trzy możliwe położenia pętli (w źródle/węźle pośrednim/ujściu), trzy rodzaje przepustowości w pętli (dodatni/ujemny/zerowy), cztery możliwe kombinacje zwrotów krawędzi, oraz cztery warianty zwielokrotnienia krawędzi w topologii sieci. Doliczenie się takiej liczby możliwych rodzajów topologii sieci, nie wymagało nawet uwzględnienia możliwości istnienia krawędzi bezpośredniej ze źródła do ujścia, czy braku którejś krawędzi. Ostatecznie zdecydowano się zatem na dziesięć różnych przypadków testowych.

### 3.1.5 Sieć z 2 węzłami pośrednimi połączonymi równolegle

W tej sekcji jako warte przetestowania wyróżniono następujące schematy sieci z 2 węzłami pośrednimi połączonymi równolegle:

- Z krawędziami o dodatniej przepustowości ze źródła do obu węzłów pośrednich i z obu węzłów pośrednich do ujścia.
- Z dodatkową krawędzią o dodatniej przepustowości z pierwszego węzła pośredniego do drugiego węzła pośredniego.
- Z dodatkową krawędzią o dodatniej przepustowości z drugiego węzła pośredniego do pierwszego węzła pośredniego.
- Z dodatkowymi krawędziami o dodatniej przepustowości z pierwszego węzła pośredniego do drugiego węzła pośredniego i z drugiego węzła pośredniego do pierwszego węzła pośredniego.<sup>1</sup>
- Z dodatkową krawędzią o dodatniej przepustowości ze źródła do ujścia.
- Z dodatkową krawędzią o dodatniej przepustowości z ujścia do źródła.
- Z dodatkową krawędzią o dodatniej przepustowości z ujścia do pierwszego węzła pośredniego.
- Z dodatkową krawędzią o dodatniej przepustowości z pierwszego węzła pośredniego do źródła.

<sup>1</sup>Potencjalne zagrożenie stworzeniem złożonej pętli.

- Z dodatkowymi krawędziami o dodatniej przepustowości z ujścia do pierwszego węzła pośredniego i z pierwszego węzła pośredniego do źródła.

Przypadki testowe dla sieci z 2 węzłami pośrednimi zostały pomyślane przede wszystkim dla przetestowania zachowania oprogramowania dla różnych schematów połączeń, stąd chociażby brak w tej sekcji klas abstrakcji z pętlami.

### 3.1.6 Sieć z 2 węzłami pośrednimi połączonymi szeregowo

W tym wypadku jako istotne przypadki testowe zdefiniowano topologie sieci z 2 węzłami pośrednimi połączonymi szeregowo:

- Z krawędziami o dodatniej przepustowości ze źródła do pierwszego węzła pośredniej, z pierwszego węzła pośredniego do drugiego węzła pośredniego i z drugiego węzła pośredniego do ujścia.
- Sieć z 2 węzłami pośrednimi połączonymi szeregowo z wieloma krawędziami pomiędzy węzłami pośrednimi.
- Sieć z 2 węzłami pośrednimi połączonymi szeregowo z krawędzią o zerowej przepustowości pomiędzy węzłami pośrednimi.
- Sieć z 2 węzłami pośrednimi połączonymi szeregowo z krawędzią o ujemnej przepustowości pomiędzy węzłami pośrednimi.
- Sieć z 2 węzłami pośrednimi połączonymi szeregowo z krawędzią z drugiego węzła pośredniego do pierwszego węzła pośredniego.<sup>2</sup>
- Sieć z 2 węzłami pośrednimi połączonymi szeregowo z krawędzią od pierwszego węzła pośredniego do ujścia.
- Sieć z 2 węzłami pośrednimi połączonymi szeregowo z krawędzią od źródła do drugiego węzła pośredniego.

Dodanie schematów z połączeniami szeregowymi zasadniczo wyczerpuje możliwości budowy schematów połączeń, każdy bardziej złożony (prawidłowy) schemat połączeń będzie bazował na pewnej kombinacji powyższych wariacji na temat połączeń szeregowych i równoległych.

### 3.1.7 Sieć z nieistniejącymi węzłami

Liczba węzłów określona w danych wejściowych jest większa niż liczba zdefiniowanych węzłów.

### 3.1.8 Sieć z ujemną liczbą węzłów

W danych wejściowych liczba węzłów w sieci jest określona za pomocą liczby ujemnej.

### 3.1.9 Sieć z $1e9$ węzłów

Test dla bardzo dużej (najlepiej granicznej) wartości liczby węzłów.

<sup>2</sup>Ujemna przepustowość pomiędzy pierwszym a drugim węzłem pośrednim nie jest tożsama z krawędzią o dodatniej przepustowości od drugiego do pierwszego węzła pośredniego. Ujemna przepustowość krawędzi jest niepoprawna z definicji, natomiast krawędź skierowana w przeciwną stronę jest jak najbardziej poprawna z punktu widzenia definicji, choć taka sieć wciąż nie miałaby żadnej przepustowości.

## 3.2 Testy biało-skrzynkowe

### 3.2.1 algs.network.VertexStructure

ToStringTest	
Id	B.1.1
Funkcja	<b>String</b> toString()
Opis	Wypisuje listę węzłów wychodzących i wchodzących.
Przypadki testowe	<ol style="list-style-type: none"><li>1. Struktura bez zdefiniowanych krawędzi wejścia/wyjścia</li><li>2. Struktura z pojedynczą krawędzią wejścia/wyjścia</li><li>3. Struktura z wieloma krawędziami wejścia wyjścia</li></ol>
Wykonawca	MO

### 3.2.2 algs.network.FlowNetworkArray

MinimalNetworkConstructorTest	
Id	B.2.1
Funkcja	FlowNetworkArray (int sourceIndex, int sinkIndex, int numVertices)
Opis	Konstruktor minimalnej struktury sieci. Inicjalizuje tylko niezbędne zmienne.
Przypadki testowe	<ol style="list-style-type: none"><li>1. Wartości oczekiwane</li><li>2. Ujemne wartości</li><li>3. sinkIndex &lt; sourceIndex</li><li>4. sinkIndex - sourceIndex != numVertices</li></ol>
Wykonawca	MO

NetworkConstructorTest	
Id	B.2.1
Funkcja	FlowNetworkArray (int sourceIndex, int sinkIndex, int numVertices), Iterator<Edges> edges
Opis	Konstruktor struktury reprezentującej graf przepływu.
Przypadki testowe	<ol style="list-style-type: none"><li>1. Wartości oczekiwane</li><li>2. Ujemne wartości</li><li>3. sinkIndex - sourceIndex != numVertices</li><li>4. sinkIndex &gt; sourceIndex</li><li>5. Pusta kolekcja krawędzi</li></ol>
Wykonawca	MO



<b>IllegalStateExceptionTest</b>	
Id	B.2.2
Funkcja	<b>void</b> validate()
Opis	<p>Metoda weryfikuje czy informacje na temat sieci są akceptowalne. Zwracany jest wyjątek, IllegalStateException w dwóch przypadkach:</p> <ol style="list-style-type: none"> <li>1. Przepływ krawędzi jest większy niż przepustowość przepustowości</li> <li>2. Ilość krawędzi wchodzących jest różna od krawędzi wychodzących</li> </ol>
Przypadki testowe	Tak jak wyżej.
Wykonawca	MO

#### **4 Realizacja testów**

#### **5 Wykonanie testów**

#### **6 Ocena rezultatów testów**

#### **7 Wnioski**

## Historia dokumentu

Data	Wersja	Autor	Szczegóły
2012-12-21	0.1.0	TC	Szablon dokumentu.
2012-12-22	0.2.0	RW	Dodanie planu testowania biało-skrzynkowego.
2012-12-22	0.2.1	MO	Dodanie planu testowania czarno-skrzynkowego.
2012-12-25	0.2.2	TC	Połączenie planów. Dodanie planu ogólnego. Rozróżnienie na plany szczegółowe.