# Pay With Friends

# Teoretyczne podstawy działania

Table of Contents

[Pay With Friends 1](#_Toc352798010)

[Teoretyczne podstawy działania 1](#_Toc352798011)

[1. Oznaczenia 2](#_Toc352798012)

[1.1. Definicje 2](#_Toc352798013)

[1.2. Przykład 2](#_Toc352798014)

[2. Użycie oznaczeń w programie 3](#_Toc352798015)

[3. Możliwe akcje użytkowników 3](#_Toc352798016)

[3.1. Ogólnie 3](#_Toc352798017)

[3.2. Dodawanie nowej transakcji 4](#_Toc352798018)

[3.3. Optymalizacja transakcji grupy użytkowników 4](#_Toc352798019)

[3.4 Globalna optymalizacja transakcji 4](#_Toc352798020)

## Oznaczenia

### Definicje

– zbiór wierzchołków grafu (wszyscy użytkownicy)

– i-ty wierzchołek (i-tyużytkownik)

– liczba użytkowników

– bilans wierzchołku i (i-tego użytkownika),

– zbiór krawędzi grafu (wszystkie połączenia)

– krawędź pomiędzy wierzchołkami „i” i „j” (przyjaźń i-tego i j-tego użytkownika), kolejność i,j nie ma znaczenia, ponieważ graf jest symetryczny

– niewypełniona płatność od użytkownika „i” do użytkownika „j”,

### Przykład



Dla sytuacji przedstawionej na obrazku:

* wierzchołki (użytkownicy):
* krawędzie (przyjaźnie):
* bilansy wierzchołków (użytkowników):
* (po zaakceptowaniu transakcji przez wszystkich użytkowników)  
  niewypełnione płatności użytkowników:

Praktyczny sens sytuacji: „użytkownik 2” jest przyjacielem użytkowników „1” i „3”. Użytkownik 2 jest winien „użytkownikowi 1” 5 jednostek, a „użytkownikowi 3” 15 jednostek.

## Użycie oznaczeń w programie

**Wierzchołek grafu – użytkownik**. Każdy z użytkowników jest identyfikowany 15 cyfrowym numerem ID (który jest identyczny co Facebookowy numer ID użytkownika).

**Krawędź grafu – przyjaźń użytkowników.** Tylko użytkownicy będący przyjaciółmi (w Pay With Friends – nie w Facebooku, tak aby użytkownik mógł decydować z kim chce mieć powiązania finansowe).

**Bilans wierzchołka –** **bilans użytkownika w transakcji.** Po dodaniu nowej transakcji dla każdego użytkownika transakcji wyliczony jest jego bilans w tej transakcji, na tej podstawie będą potem wyliczone płatności pomiędzy użytkownikami.

## Możliwe akcje użytkowników

### 3.1. Ogólnie

**Stworzenie użytkownika** – stworzenie wierzchołka reprezentującego daną osobę w aplikacji. Zarówno złożoność obliczeniowa jak i pamięciowa tej operacji wynosi .

**Dodanie/potwierdzenia przyjaźni** – stworzenie krawędzi pomiędzy dwoma użytkownikami w aplikacji. Zarówno złożoność obliczeniowa jak i pamięciowa tej operacji wynosi .

**Dodanie nowej transakcji** – zmiana bilansów pewnej grupu użytkowników (która będzie miała miejce dopiero po zaakceptiowaniu transakcji przez wszystkich użytkowników). Bilansy zostaną od razu przekształcone w płatności pomiędzy użytkownikami (jeśli nie jest to możliwe(brak krawędzi pomiędzy użytkownikami) transakcja nie będzie mogła być zaakceptowana).

**Optymalizacja transakcji grupy użytkowników** – zmiana płatności pomiędzy pewną grupą wierzchołków tak, aby zminimalizować pewien wskaźnik transakcji (np. ilość niezerowych płatności, suma wartości płatności).

**Globalna optymalizacja transakcji** – analogiczna optymalizacja przeprowadzona pomiędzy wszystkimi wierzchołkami.

### 3.2. Dodawanie nowej transakcji

Akcje potrzebne do dodania nowej transakcji (n – liczba uczestników transakcji):

* obliczenie bilansu każdego użytkownika (proste działania artymetyczne) – złożoność
* stworzenie MSP (Minimal Spanning Tree) minimalnego drzewa spinającego wszystkich użytkowników po dostępnych krawędziach (przy zastosowaniu algorytmu Prima złożoność )– jeśli nie jest możlwe stworzenie MSP, oznacza to, że transakcja nie może zostać wprowadzona do systemu – potrzebne są dodatkowe krawędzie – przyjaźnie pomiędzy użytkownikami
* dla każdego wierzchołka mającego 1 krawędź stworzenie transakcji zmniejszającej bilans tego użytkownika do 0 i usunięcie krawędzi i wierzchołka z MSP (złożoność

Cały algorytm ma więc złożoność obliczeniową równą

### 3.3. Optymalizacja transakcji grupy użytkowników

W celu optymalizacja transakcji w grupie użytkowników korzystam z identycznego algorytmu co przy dodawaniu nowej transakcji, lecz jako bilans wierzchołka obliczam jego bilans względem grupy wierzchołków w obrębie której odbywa się transakcja.  
Złożoność algorytmu wynosi więc

### 3.4 Globalna optymalizacja transakcji

W celu optymalizacji globalnej można zastosować identyczny algorytm co przy lokalnej optymalizacji transakcji – wtedy algorytm ma złożoność . Przy bardzo dużej liczbie wierzchołków. Taka złożnoność może się okazać niewystarczająca, więc warto zastanowić się nad możliwością optymalizacji tego algorytmu.