

Analiza sieci społecznościowych w narzędziu Pajek (część I)

Celem pierwszej części ćwiczenia jest zapoznanie się z narzędziem Pajek i wykonanie prostych analiz przy jego wykorzystaniu. Pajek to darmowy program do analizy sieci społecznościowych, przygotowany i napisany przez V.Batagelja i A.Mrvara z Uniwersytetu w Ljubljanie (Słowenia). Program jest dostępny pod adresem <http://vlado.fmf.uni-lj.si/pub/networks/pajek/>. Wraz z programem dostępne są:

- strona Wiki (<http://pajek.imfm.si/doku.php>)
- podręcznik użytkownika (<http://vlado.fmf.uni-lj.si/pub/networks/pajek/doc/pajekman.pdf>)
- dokumentacja (<http://vlado.fmf.uni-lj.si/pub/networks/doc/>)

Reprezentacja sieci

Podstawowym formatem zapisywania grafów jest plik *.net. Można w nim zapisać dane w następujących formatach:

- lista krawędzi *Edges
- lista łuków *Arcs
- lista incydencji *Arcslist
- macierz sąsiedztwa *Matrix

Oprócz pliku *.net Pajek korzysta jeszcze z następujących typów plików:

- *.clu : pliki z etykietami wierzchołków, służą przede wszystkim do definiowania partycji w zbiorach wierzchołków
- *.vec : pliki z danymi ilościowymi opisującymi wierzchołki
- *.per : permutacje, czyli uszeregowania wierzchołków

Wszystkie dane o przetwarzanej sieci mogą być umieszczone w pojedynczym pliku *.paj, wewnątrz którego pojawiają się sekcje reprezentujące poszczególne komponenty sieci. Sekcje rozpoczynają się zawsze od nagłówków:

- Network moja_siec (zawartość pliku *.net)
- Partition moja_partycja (zawartość pliku *.par)
- Permutation moj_porzadek (zawartość pliku *.per)
- Vector moj_wektor (zawartość pliku *.ver)

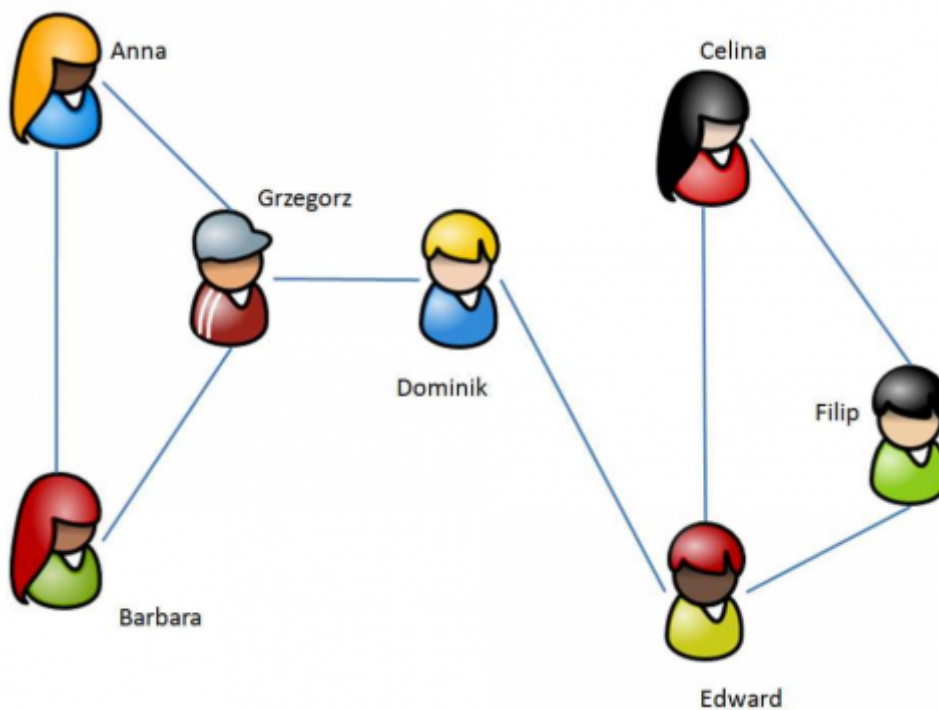
Pojedynczy plik projektu może zawierać wiele sekcji tego samego typu. Poniżej znajduje się przykład pliku projektu.

```
*Network Pan network
*Vertices 5
1 "A"
2 "B"
3 "C"
4 "D"
5 "E"
*Edges
1 2 1
```

```
2 3 1
2 4 1
3 5 1
4 5 1
*Partition Tail&Head
*Vertices 5
1
2
2
2
2
```

Praca z grafem nieskierowanym

1. Utwórz plik tekstowy `friends.net` i umieść w nim model następującej rzeczywistości (związek między osobami reprezentuje znajomość).



2. Załaduj utworzony plik do programu Pajek i narysuj graf, wybierając z menu opcję `Draw→Network` (lub korzystając ze skrótu klawiaturowego `Ctrl+G`)

3. Przeczytaj najbardziej podstawowe statystyki dotyczące załadowanej sieci. W tym celu skorzystaj z `Network→Info→General` (lub korzystając ze skrótu klawiaturowego `F7`) i pozostaw wartość 0 w oknie dialogowym.

4. Wyznacz stopnie wszystkich wierzchołków, tworząc nową partycję i przypisując każdy wierzchołek do właściwej partycji. Wybierz `Network→Create Partition→Degree→All`, a następnie otwórz edytor partycji i sprawdź wyznaczone wartości. Wyświetl partycję za pomocą `Draw→Network+First Partition` (lub skorzystaj ze skrótu klawiaturowego `Ctrl+P`). W oknie z wizualizacją grafu zmień algorytm rysowania grafu wybierając `Layout→Energy→Kamada-Kawai→Separate Components` (lub wykorzystaj skrót `Ctrl+K`)

5. Wyznacz najkrótszą drogę między Barbarą i Filipem. W tym celu utworzysz nową sieć, na którą

będą składały się jedynie wierzchołki należące do znalezionej ścieżki. Dodatkowo, utworzysz partycję dzielącą oryginalną sieć na wierzchołki należące i nienależące do najkrótszej ścieżki między Barbarą i Filipem. Wybierz `Network→Create New Network→Subnetwork with Paths→One Shortest Path Between Two Vertices`. W pierwszym oknie dialogowym wprowadź „Barbara”, w drugim oknie wprowadź „Filip”. Odpowiedz twierdząco na pytania o porzuceniu wartości na krawędziach i identyfikacji węzłów na ścieżce. Wyświetl uzyskaną sieć wykorzystując opcję `Draw→Network`.

6. W sekcji *Networks* powróć do oryginalnej sieci, składającej się z 7 wierzchołków. Narysuj partycję dzielące wierzchołki na te, które znajdują się na ścieżce z punktu 5, i te poza ścieżką.

7. Wyznacz miarę bliskości dla wszystkich węzłów, korzystając z menu `Network→Create Vector→Centrality→Closeness→All`. Obejrzyj uzyskany wektor, klikając na przycisk edycji. Zauważ, że wyznaczone wartości mieszczą się w przedziale $[0;1]$. Przemnóż wartości w wektorze przez 100 (`Vector→Transform→Multiply By` i w oknie dialogowym wpisz 100). Narysuj sieć, uzależniając wielkość wierzchołka od wyznaczonej miary bliskości (`Draw→Network+First Vector`).

8. Podziel wierzchołki na dwa zbiory reprezentujące niską i wysoką miarę bliskości. W tym celu wybierz `Vector→Make Partition→by Intervals→Selected Thresholds` i w oknie dialogowym wprowadź „#2” (zauważ, że możesz także samodzielnie zdefiniować granice partycji). Wyświetl uzyskane partycje.

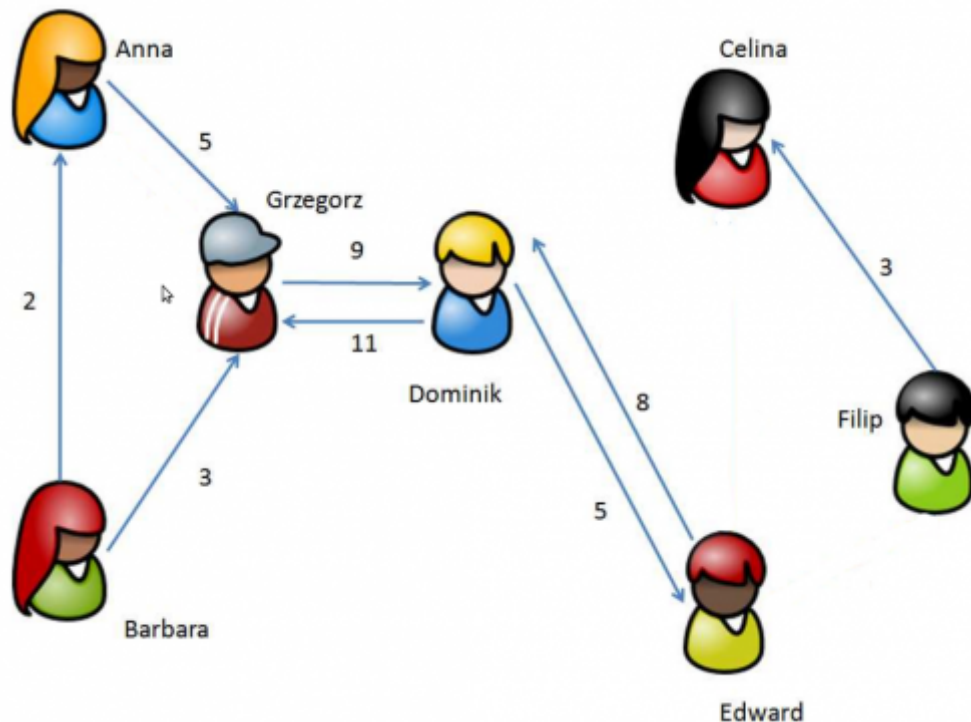
9. Dla każdego wierzchołka wylicz miarę pośrednictwa (`Network→Create Vector→Centrality→Betweenness`). Obejrzyj uzyskane wyniki i sprawdź, czy zgadzają się z Twoją intuicją. Narysuj sieć (`Ctrl+G`) i kliknij prawym klawiszem myszy na wierzchołku „Anna”. Kliknij dwukrotnie na słowie *Newline* i w oknie dialogowym wpisz „6” (tworzysz w ten sposób nową krawędź między Anną i Filipem). W oknie z wizualizacją sieci wybierz z menu opcję `Redraw`. Ponownie wyznacz miarę pośrednictwa dla wierzchołków i zinterpretuj uzyskany wynik.

10. Znajdź bliższych i dalszych znajomych Barbary. Wybierz `Network→Create Partition→k-Neighbours→All`, w oknie dialogowym wpisz „Barbara”, w następnym oknie dialogowym podaj odległość „2”. Narysuj uzyskaną partycję (`Ctrl+P`).

11. Utwórz plik `friends.sex.clu` zawierający partycję dzielącą wierzchołki na kobiety i mężczyzn. Wczytaj plik do programu Pajek korzystając z opcji `File→Partition→Read` i wyświetl partycję.

Praca z grafem skierowanym

12. Zamień sieć z poprzedniego ćwiczenia na poniższą sieć, w której krawędzie między węzłami reprezentują wysyłanie wiadomości e-mail, a wartości krawędzi reprezentują liczbę wymienionych wiadomości. Możesz wybrać dowolny format danych wejściowych rozpoznawanych przez Pajek. Jeśli jesteś leniwa/y, posłuż się plikiem [friends.direct.net](https://semantic.cs.put.poznan.pl/wiki/TSISS/)



13. Wyświetl stworzoną przez siebie sieć. Upewnij się, że wyświetlane są zarówno etykiety wierzchołków (Options→Mark Vertices Using→Labels, Ctrl+L) jak i wagi krawędzi (Options→Lines→Mark Lines→with Values, Ctrl+V)

14. Wyznacz stopnie wejściowe i wyjściowe wszystkich wierzchołków, zapisz wyznaczone wartości w postaci dwóch wektorów (Network→Create Vector→Centrality→Degree→Input, Output). Następnie w sekcji Vectors w pierwszym polu wybierz wektor o nazwie Input Degree of Nx (7), a w drugim polu wybierz wektor o nazwie Output Degree of Nx (7). Twój ekran powinien wyglądać jak poniżej.



15. Na bazie dwóch aktualnie wybranych wektorów wyznacz nowy wektor, którego wartości będą odpowiadały stosunkowi stopnia wejściowego i wyjściowego. Wybierz z menu opcję Vectors→Divide i wyświetl uzyskany wynik (Draw→Network+First Vector).

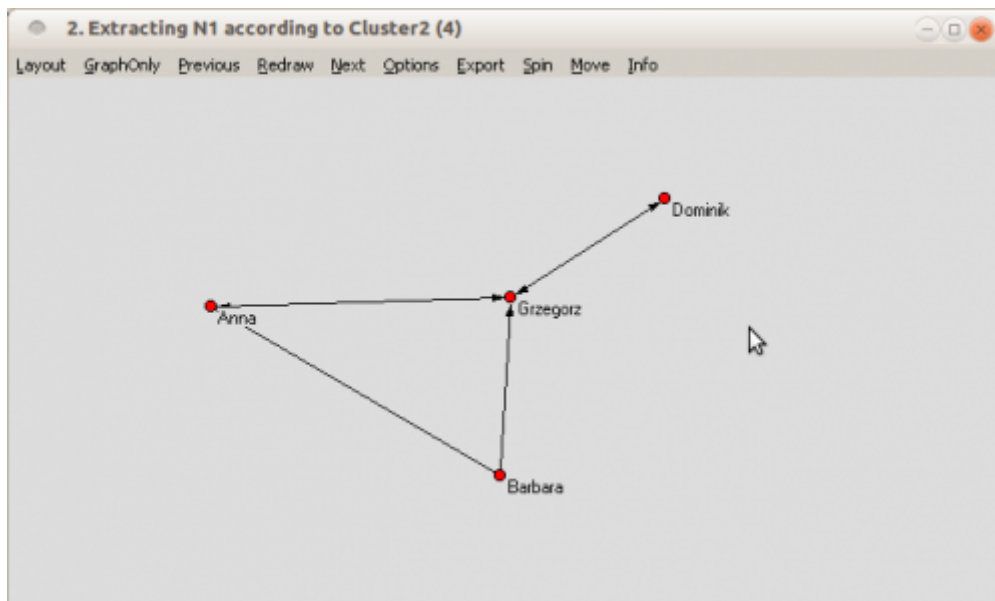
16. Znajdź silnie spójne komponenty sieci wykorzystując w tym celu opcję Network→Create Partition→Components→Strong (jako minimalny rozmiar partycji podaj „1”), a następnie wyświetl znalezione partycje. W analogiczny sposób znajdź słabo spójne komponenty sieci i wyświetl uzyskany wynik. Czy potrafisz wytłumaczyć różnicę? Dodaj do sieci jedną krawędź w taki sposób, aby liczba silnie spójnych komponentów sieci wyniosła 3.

17. Wyznacz miarę centralności węzłów wg. bliskości (weź pod uwagę krawędzie wychodzące). Kogo należy powiadomić, aby informacja rozniosła się między przyjaciółmi najszybciej?

18. Wylicz najkrótsze ścieżki między wszystkimi parami wierzchołków. Wybierz Network→Create New Network→SubNetwork with Paths→Geodesics Matrices. Zauważ, że ta operacja tworzy dwie nowe sieci, nazwane Geodesics Count Matrix of Nx (7) oraz Shortest Path Length Matrix of Nx (7). Wyświetl tę drugą sieć i zaobserwuj, ile wynoszą odległości między wierzchołkami, które należą do różnych składowych spójności sieci. Wyświetl średnicę sieci

(Network→Create New Network→SubNetwork with Paths→Info on Diameter).

19. Wyznacz wszystkie ścieżki o długości co najwyżej 3 prowadzące od Barbary. W tym celu posłuż się opcją Network→Create Partition→k-Neighbours→Output, wskaż jako wierzchołek początkowy wierzchołek o etykiecie „Barbara” i podaj „2” jako maksymalną długość ścieżki. Czy Edward należy do najbliższych znajomych Barbary? Następnie utwórz podsieć składającą się z najbliższych znajomych Barbary. W pierwszym kroku zapisz wszystkie wierzchołki z partycji 0,1 i 2 (Barbara, jej przyjaciele, przyjaciele jej przyjaciół) do klastra korzystając z operacji Partition→Make Cluster→Vertices from Selected Clusters i podając w oknie parametrów wartość „0-2”. Następnie, stwórz nową sieć korzystając z opcji Operations→Network+Cluster→Extract Subnetwork. Twoja końcowa sieć powinna wyglądać następująco:



zadanie samodzielne

Pobierz dane potrzebne do wykonania ćwiczenia (dostarczone przez prowadzącego) i wypakuj z archiwum plik [SanJuanSur2.paj](#). Załaduj projekt do programu Pajek korzystając z opcji File→Pajek Project File→Read. Plik zawiera informacje o związkach między 75 rodzinami wiejskimi z miejscowości [San Juan Sur w Kostaryce](#), informacje zebrano w 1948 roku. Łuki reprezentują częste wizyty domowe (łuk prowadzi od rodziny wizytującej do rodziny goszczącej, dokładna liczba wizyt nie została odnotowana), przy czym łuki posiadają następujące wagi:

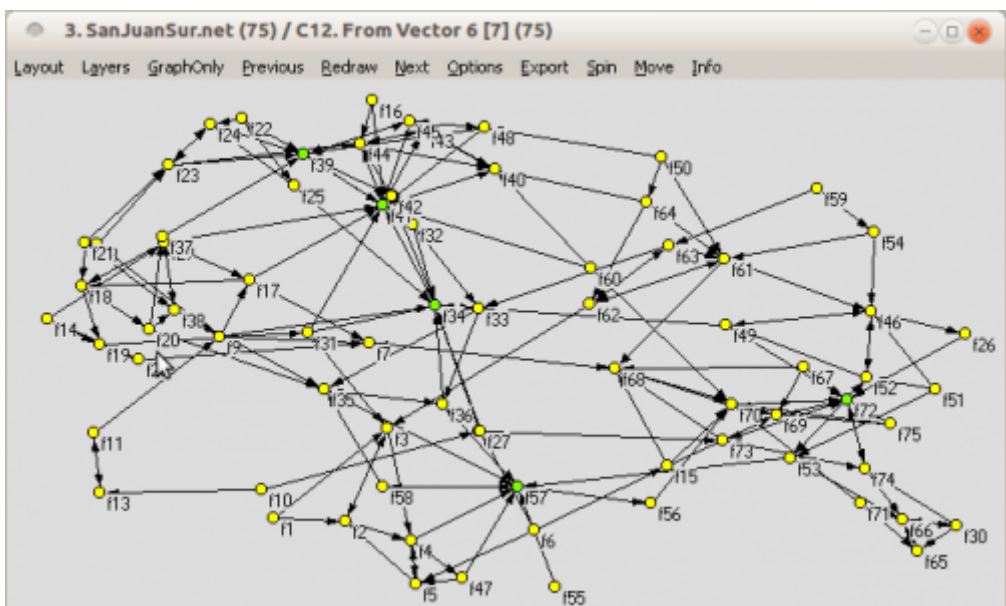
- (1) wizyta sąsiedzka
- (2) wizyta rodzinna
- (3) wizyta rytualna (związana z jakimś obrzędem religijnym)

W zbiorze występują też dwie partycje:

- SanJuanSur_status: przypisanie każdej z rodzin do jednej z 14 klas na podstawie statusu socjo-ekonomicznego, większa liczba oznacza bogatszą i ważniejszą rodzinę
- SanJuanSur_leaders: binarna flaga reprezentująca fakt występowania w rodzinie liderki/lidera, tj. osoby cieszącej się w społeczności szczególną estymą

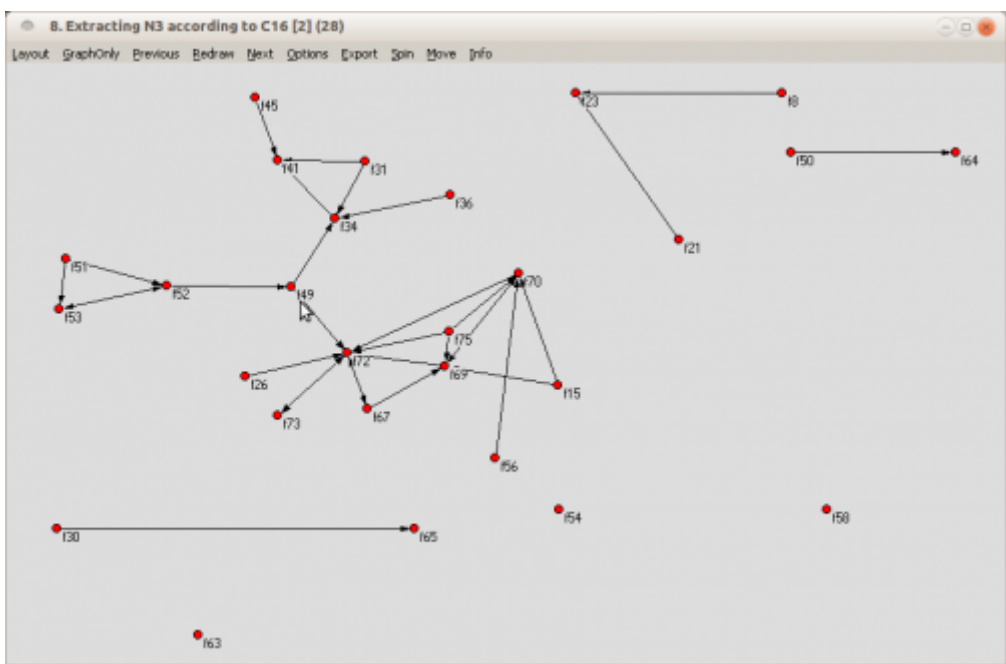
1. Wyświetl sieć i postaraj się dobrać optymalny algorytm rozmieszczenia wierzchołków

2. Dla każdego wierzchołka wyznacz stopień wejściowy wierzchołka (ile rodzin odwiedziło daną rodzinę) i zidentyfikuj pięć rodzin, które posiadają najwyższy stopień wejściowy. Następnie podziel wierzchołki na partycje w taki sposób, aby w pierwszej partycji znalazło się pięć najważniejszych rodzin, a pozostałe rodziny trafiły do drugiej partycji. Narysuj wynikową sieć z zaznaczonymi partycjami.

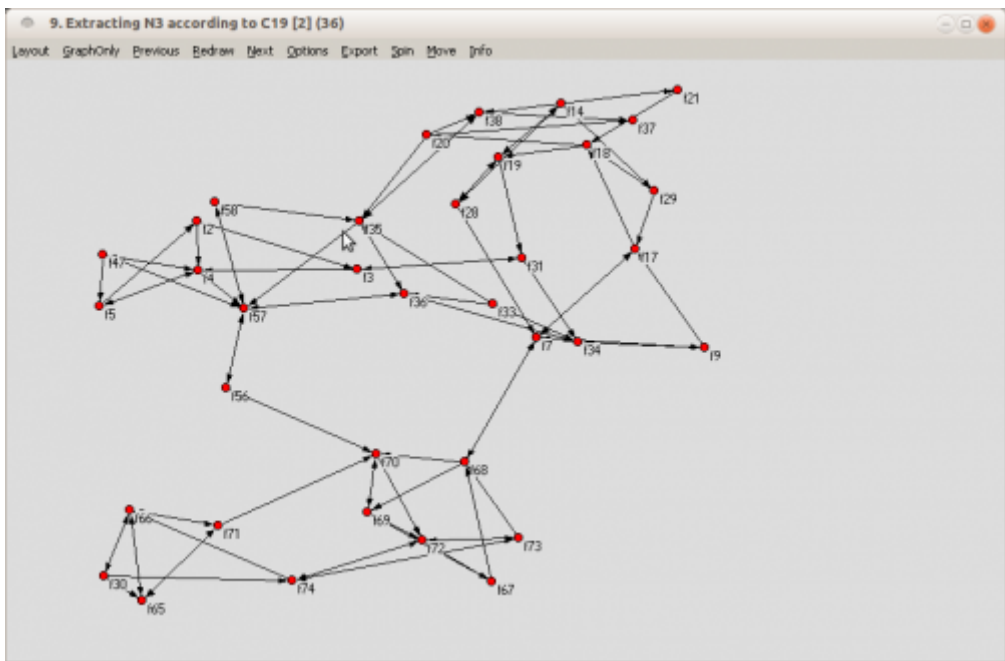


3. Załóżmy, że plotka potrafi propagować się przez dwa spotkania (tzn. jeśli plotka pojawi się w trakcie wizyty rodziny A u rodziny B, to B przekaze plotkę rodzinie C, a rodzina C może przekazać tę plotkę rodzinie D). Załóżmy, że źródłem plotki jest rodzina f21. Ile rodzin dowie się o plotce, jeśli plotka jest powielana w trakcie każdej wizyty?

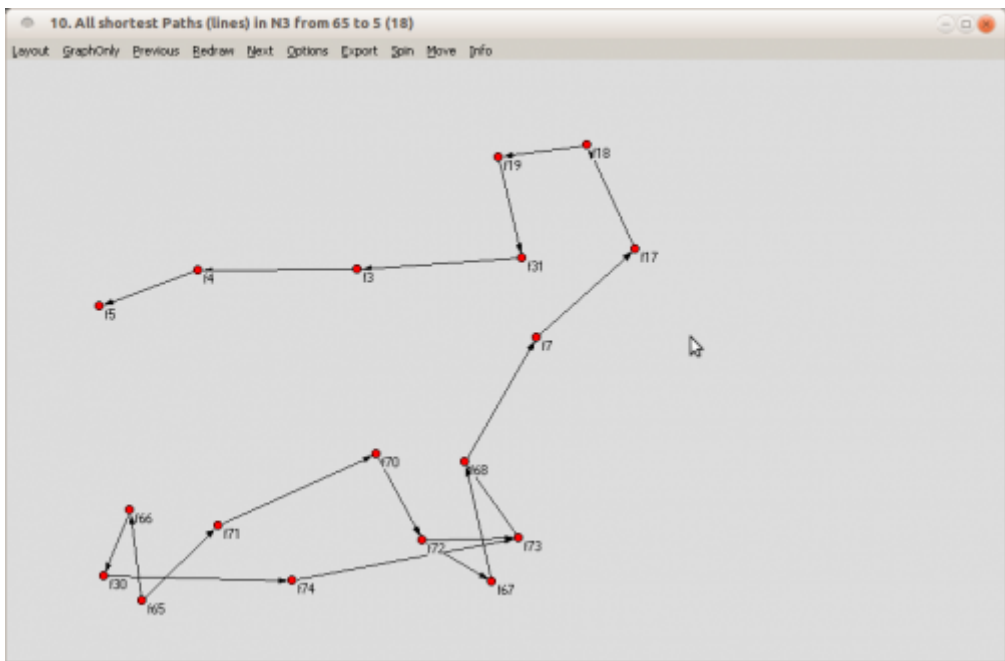
4. Wykorzystaj partycję SanJuanSur_status i na jej podstawie stwórz nowy podział zbioru wierzchołków na rodziny biedne (status [1-5]), rodziny średniozamożne (status [6-10]) i rodziny bogate (status [11-14]). Na bazie oryginalnej sieci zbuduj nową sieć prezentującą jedynie związki między rodzinami średniozamożnymi.



5. Znajdź największy silnie spójny komponent sieci. Ile wynosi średnica tego komponentu?



6. Załóżmy, że rodzina f65 pragnie złożyć propozycję matrymonialną rodzinie f5. Wyświetl wszystkie ścieżki, jakimi ta propozycja trafi do adresatów. Ile jest potrzebnych wizyt „po drodze”?



From: <https://semantic.cs.put.poznan.pl/wiki/TSiSS/> - Technologie semantyczne i sieci społecznościowe

Permanent link: <https://semantic.cs.put.poznan.pl/wiki/TSiSS/doku.php?id=laboratorium-pajek-1>

Last update: 2017/04/19 02:51

