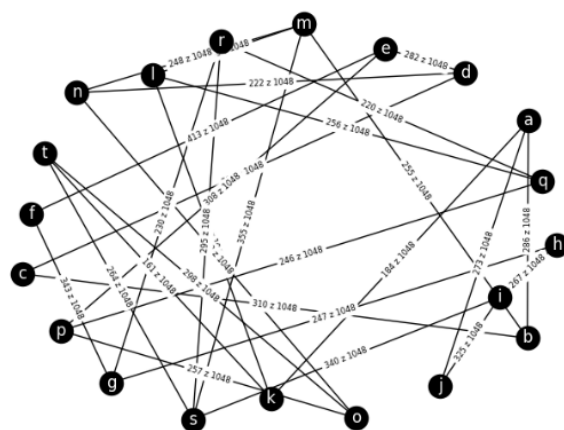
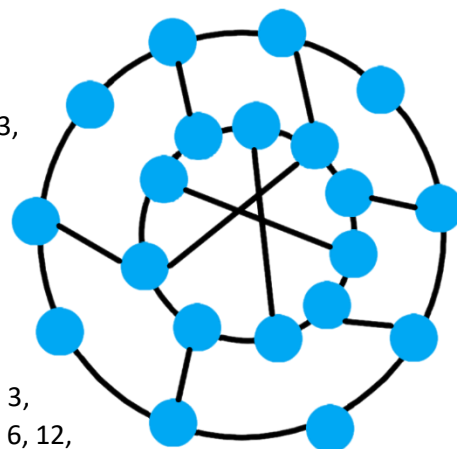


Sprawozdanie nr 2

- Topologia G (w programie jest to graf 1 (-d 1), rysunki obok):

- N może być dowolne, np.:

[[0, 13, 13, 15, 12, 8, 13, 15, 10, 3, 10, 5, 2, 5, 4, 10, 13, 0, 2, 5], [13, 0, 3, 1, 4, 3, 10, 2, 7, 13, 0, 4, 2, 13, 8, 1, 0, 1, 10, 6], [3, 6, 0, 13, 8, 15, 15, 12, 10, 0, 8, 10, 11, 10, 15, 14, 10, 6, 2, 14], [13, 2, 7, 0, 8, 9, 10, 8, 15, 9, 2, 9, 2, 11, 10, 11, 10, 15, 6, 5], [14, 15, 14, 15, 0, 5, 13, 4, 8, 15, 13, 7, 5, 10, 9, 5, 3, 0, 2, 9], [7, 15, 2, 2, 9, 0, 7, 0, 3, 4, 3, 10, 3, 2, 11, 6, 11, 15, 6, 10], [15, 15, 6, 3, 13, 0, 0, 3, 1, 1, 2, 2, 2, 10, 4, 8, 1, 9, 0, 13], [1, 6, 5, 8, 5, 15, 4, 0, 6, 15, 4, 0, 11, 7, 9, 0, 4, 4, 5, 5], [2, 8, 3, 14, 3, 0, 2, 5, 0, 8, 2, 10, 13, 11, 14, 5, 15, 14, 8, 5], [1, 12, 5, 5, 2, 3, 14, 3, 0, 0, 14, 3, 12, 7, 14, 15, 15, 2, 6, 14], [8, 5, 1, 5, 2, 2, 13, 6, 10, 2, 0, 12, 6, 12, 4, 0, 4, 10, 4, 2], [1, 14, 1, 2, 0, 11, 10, 6, 3, 0, 5, 0, 12, 12, 0, 3, 4, 14, 9, 7], [11, 3, 9, 8, 13, 13, 8, 5, 15, 14, 10, 3, 0, 11, 9, 2, 4, 4, 9, 0], [15, 4, 14, 1, 15, 13, 12, 15, 4, 9, 5, 4, 8, 0, 1, 7, 13, 10, 8, 5], [0, 3, 1, 15, 9, 15, 6, 10, 0, 12, 8, 10, 4, 7, 0, 13, 13, 6, 6, 4], [10, 6, 9, 14, 13, 1, 3, 14, 13, 6, 10, 6, 3, 4, 4, 0, 15, 11, 9, 2], [8, 9, 1, 1, 9, 10, 6, 9, 9, 13, 10, 9, 15, 9, 5, 1, 0, 11, 5, 10], [2, 8, 4, 15, 13, 7, 15, 2, 6, 4, 9, 9, 14, 10, 1, 4, 10, 0, 0, 2], [5, 8, 10, 8, 13, 11, 4, 2, 2, 0, 8, 2, 12, 1, 4, 3, 15, 8, 0, 8], [1, 1, 4, 14, 13, 2, 13, 9, 1, 8, 9, 2, 11, 15, 6, 10, 11, 6, 7, 0]]



- Funkcja przepustowości:
 - Przepustowość generowana jest jako podłoga z $4 \cdot$ średnia $a(e)$ (w przypadku wybrania generowania grafu z pliku, przepustowość należy podać ręcznie, chyba że występuje flaga -ac)
 - jest to połączenie `generate_a(first_time=True)` z `generate_c()`
- Funkcja przepływu:
 - Funkcja, która dla każdego elementu z siatki natężeń (za wyjątkiem gdy $i=j$) odnajduje najszybszą drogę.
 - Występuje w kilku wersjach:
 - `generate_a(quick=True)` – odmiana w której brane pod uwagę są połączenia pozwalające jedynie na pełne przesłanie pakietów
 - (default) `generate_a(quick=False)` – odmiana w której brane pod uwagę są wszystkie połączenia umożliwiające przesłanie co najmniej 1 pakietu, czyli że 40 pakietów będzie wysłane maksymalnie przez 40 oddzielnych dróg, pod warunkiem że w żadnej z dotychczasowych wybranych dróg nie dało się więcej wysłać - zalecana
 - `generate_a(first_time=True)` – nie brana jest pod uwagę zależność $c(e) > a(e)$, wykorzystuje algorytm 2
 - `generate_a(first_time=True, quick=True)` – nie brana jest pod uwagę zależność $c(e) > a(e)$, wykorzystuje algorytm 1 – zalecana

- Program przyjmuje wiele parametrów:

- -h wyświetla pomoc
- -g nazwapliku/1/2 ładuje do programu graf z pliku bądź bierze predefiniowany
 - struktura pliku do odczytania:
 - 1 linia: rozpisana po kolei macierz
 - 2 linia: nazwy wierzchołków (muszą być wszystkie nazwy występujące)
 - następne linie: początek krawędzi, koniec, współczynnik p, współczynnik m, współczynnik c
 - Przykładowy graf:

0 3 2 3 9 12 0 11 6 10 12 5 0 10 6 12 11 2 0 6 7 12 15 14 0

q w e r t

q e 0.5 30 80

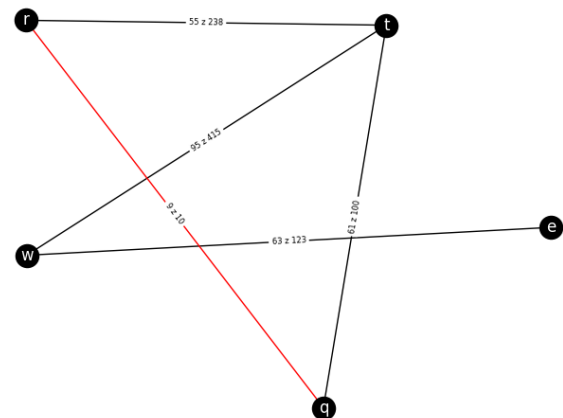
q r 0.4 13 50

r t 0.9 42 10000

w e 0.99 1 123

w t 0.99 3 1245

q t 0.5 10 1000



- -e numer zadania(kropki)
- -t maksymalne opóźnienie pakietu, domyślnie jest 1
- -p prawdopodobieństwo działania krawędzi (nadpisuje domyślne ustawienia w grafie z pliku) (domyślnie dla wbudowanych grafów jest 0.95)
- -ac generuje c(e) dla grafu z pliku zgodnie ze wzorem powyżej
- -m ustawia globalnie średnią wielkość pakietu
 - Przykładowe uruchomienie dla grafu 1:

```

[artur@artur-virtualbox ~/PycharmProjects/grafy]$ ./mygraph.py -g 1 -e 2 -p 0.8
Network reliability t= 0.013579928715919516 59.589999999999996 %
  
```

- Uwaga! – mogą występować nieznaczące odchylenia danych, spowodowane stosunkowo małą ilością prób (10000 z powodu czasu działania programu)

- Zwiększanie wartości w macierzy natężeń

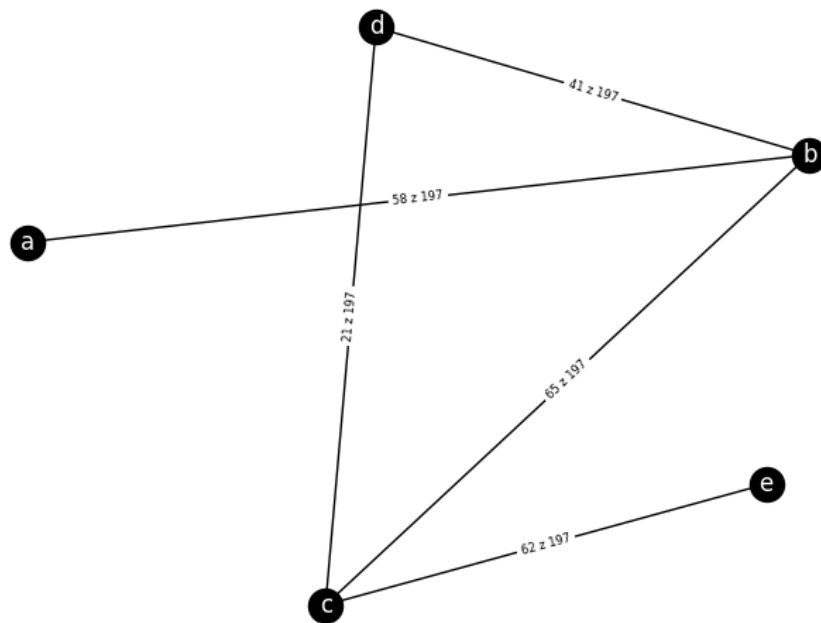
- (dla grafu z punktu 1 (p=0.95))

```

Every iteration, N is multiplied by 1.10
0 t= 0.0038787834202788987 98.48 %
1 t= 0.005671635616225128 98.14 %
2 t= 0.00918679988719226 97.34 %
3 t= 0.01920714939685744 96.46000000000001 %
4 t= 0.03961424608538123 95.59 %
5 t= 0.08138160107523847 90.14 %
6 t= 0.14149600072317164 79.46 %
7 t= 0.16149363103822453 58.78 %
8 t= 0.1457402322013568 31.990000000000002 %
9 t= 0.0 0.0 %
  
```

- Początkowa sprawność sieci sięga aż do 98,5%
- Wraz ze wzrostem natężenia, powoli spada, by przy 6 powiększeniu znacząco spadła o ponad 10 procent
- W następnych iteracjach sieć lawino traci na wydajności, by spadła do 0 dla 9

- (dla grafu nr 2)



```
[artur@artur-virtualbox ~/PycharmProjects/grafy]$ ./mygraph.py -g 2 -e 3 -t 10
```

Every iteration, N is multiplied by 1.10

```

0 t= 0.009095050109572457 89.39 %
1 t= 0.011283604288086916 89.87 %
2 t= 0.014092360647863686 89.68 %
3 t= 0.01753550726675047 89.82 %
4 t= 0.022921428160629684 89.56 %
5 t= 0.03218551453782535 89.51 %
6 t= 0.08028388543157139 81.78 %
7 t= 0.04109220966445936 76.44999999999999 %
8 t= 0.06722482073791762 77.18 %
9 t= 0.22483591144163934 76.44 %
10 t= 0.0 0.0 %
  
```

- Początkowa sprawność sieci jest w okolicach 90%, sieć działa dosyć sprawnie do czasu, gdy nie jest w stanie spełnić norm przepustowości i nagle, przy 9 zwiększeniu macierzy natężeń drastycznie spada niezawodność z 76% do 0% (sieć jest w stanie udźwignąć jedynie 2 krotne zwiększenie natężenia)

- Zwiększenie przepustowości:
 - Dla grafu 1 oraz 2 operacja nie miała wpływu, ponieważ grafy te są na tyle dobrze zbudowane, że największy wpływ na wynik ma p
 - Dla grafu z pliku tekstowego:


```

0 3 2 3 9 12 0 11 6 10 12 5 0 10 6 12 11 2 0 6 7 12 15 14 0
w q e r t
q e 0.5 30 80
q r 0.4 5 50
r t 0.9 4 100
w e 0.79 1 123
          
```

```

Every iteration, c(e) is multiplied by 1.10
0 t= 0.05899700499585441 35.709999999999994 %
1 t= 0.018615525754857466 35.31 %
2 t= 0.011226477623830904 35.949999999999996 %
3 t= 0.01052052951892616 37.01 %
4 t= 0.01721306472918303 50.580000000000005 %
5 t= 0.010284117832813463 51.32 %
6 t= 0.007334448810488445 50.91 %
7 t= 0.017701358487264493 55.08 %
8 t= 0.00685819893513286 55.1 %
9 t= 0.004929868015227514 54.069999999999999 %
10 t= 0.0040332541480955055 54.36 %
11 t= 0.0034875074292494567 54.97 %
12 t= 0.0029708292107833495 55.059999999999995 %
13 t= 0.0025631737387304654 54.76 %
14 t= 0.0022189965981111582 54.13 %
15 t= 0.00196774460920575 54.890000000000001 %
16 t= 0.0017161414069438944 54.6 %
17 t= 0.0015224180500397146 54.61 %
18 t= 0.0013753902499198162 55.69 %
19 t= 0.0012031252500799877 54.32 %
20 t= 0.0010795328853660136 55.05 %
21 t= 0.0009537536083300427 53.779999999999994 %
22 t= 0.0008730028466279074 55.16 %
23 t= 0.0007711452528616342 54.15 %
24 t= 0.0006960407576158288 54.59 %
25 t= 0.000635494693468726 55.179999999999999 %
26 t= 0.0005634726079790123 54.47 %
27 t= 0.0005064545767126689 54.32 %
28 t= 0.00046161077235407826 55.03 %
29 t= 0.0004121284636073919 54.190000000000005 %
30 t= 0.000377435312875059 54.96 %
31 t= 0.00033411887611811 53.959999999999994 %
32 t= 0.00030889797318419185 54.900000000000006 %
33 t= 0.00027943258352601454 54.879999999999995 %
34 t= 0.0002558665966043269 55.669999999999995 %
35 t= 0.00022867343941700038 55.1 %
36 t= 0.00020740062865601402 54.7 %
37 t= 0.00018414116206569742 53.94 %
38 t= 0.00016934708062227614 54.49 %

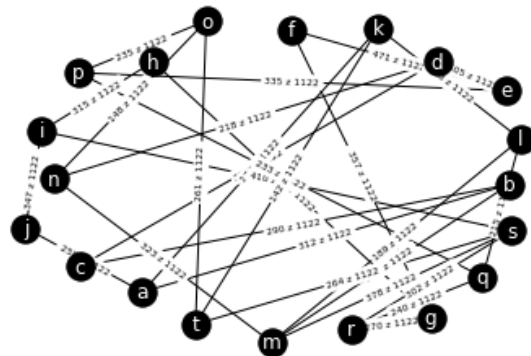
```

- Znacząco spadło opóźnienie sieci oraz nastąpiła zmiana sprawności sieci z 35% do 54%
- Należy zauważyć, że dalsze powiększanie przepustowości mija się z celem, ponieważ sprawność sieci zależy już wyłącznie od sprawności krawędzi tworzących graf

- Dodawanie nowych krawędzi:

- Graf 1

Every iteration, new edge is added to the graph



```

0 t= 0.011288099251645803 60.01 %
n s
1 t= 0.04238079348903055 62.19 %
n r
2 t= 0.0625550375320599 62.09 %
m o
3 t= 0.07169288738783218 63.739999999999995 %
g p
4 t= 0.08197784742349283 65.81 %
e o
5 t= 0.11115274249700176 66.24 %
b d
6 t= 0.12660766196307677 68.01 %
h s
7 t= 0.15122077672930817 69.17 %
a s
8 t= 0.1844246136541687 69.08999999999999 %
b i
9 t= 0.20432453935350525 70.14 %
b g
10 t= 0.21173919691030538 71.39 %
a t
11 t= 0.20270026237353214 71.94 %
l o
12 t= 0.19451938035353122 71.75 %
b k
13 t= 0.20076833660731905 72.03 %
o q
14 t= 0.19827226167775114 72.1 %
d i
15 t= 0.15830696040830466 74.02 %
h n
16 t= 0.14143922197804779 76.29 %
c p
17 t= 0.1180215880945356 76.88000000000001 %

```

- Dodanie nowych krawędzi powoli zwiększa sprawność sieci, jednakże z każdą kolejną krawędzią widać, że średnie t znacząco rośnie (tak jest do 11, potem następuje stabilizacja)

▪

- Graf 2

```

/home/artur/PycharmProjects/grafy/venv/bin/python /home/artur/PycharmProjects/grafy/mygraph.py -g 2 -e 5
Every iteration, new edge is added to the graph
0 t= 0.0100912452134309 89.42 %
b e
1 t= 0.014542735670196623 89.2 %
d e
2 t= 0.029010383977591515 93.91000000000001 %
a c
3 t= 0.031616928017536586 94.07 %
a e
4 t= 0.02140342270568448 99.24 %
a d
5 t= 0.018390118938056664 99.87 %

```

- Dodanie nowych krawędzi znacząco wpłynęło na sprawność sieci
- Należy zauważyć że zwiększenie sprawności sieci ma też taką konsekwencję, że wzrosło średnie t. Spowodowane to jest możliwością występowania ekstremalnych przypadków, oraz nowe krawędzie posiadają stosunkowo małą przepustowość
- Graf z pliku tekstowego z wykorzystaniem flagi -ac (automatycznie generowane c(e)):

```
0 3 2 3 9 12 0 11 6 10 12 5 0 10 6 12 11 2 0 6 7 12 15 14 0
```

```
a b c d e
```

```
a b 0.75 1 100
```

```
b c 0.75 1 100
```

```
c d 0.75 1 100
```

```
d e 0.75 1 100
```

```

/home/artur/PycharmProjects/grafy/venv/bin/python /home/artur/PycharmProjects/grafy/mygraph.py -g s.txt -
Every iteration, new edge is added to the graph
0 t= 0.002528030233840032 31.290000000000003 %
b e
1 t= 0.010158014632176677 39.62 %
a c
2 t= 0.01651802713848852 58.309999999999995 %
c e
3 t= 0.018262828957095547 69.83 %
a e
4 t= 0.016909004543828605 82.39 %
a d
5 t= 0.018468162653011877 92.16 %
b d
6 t= 0.01865998449119707 96.23 %

```

- Dodanie nowych krawędzi znacząco zwiększyło sprawność, niestety nowe krawędzie posiadają mniejszą przepustowość, co za tym idzie spowalniają sieć
- Spostrzeżenia:
 - Kluczowe jest dobre zaprojektowanie grafu
 - Największą rolę ma przepustowość, dobrze jest gdy jest kilkakrotnie większa od aktualnego przepływu, tak by w razie awarii była w stanie zapewnić dalszą integralność sieci
 - Ważne jest by połączenia miały jak największą sprawność, ponieważ zniszczona krawędź jest dużym obciążeniem dla sieci

- Chcąc zachować jak największą szybkość sieci nie należy dodawać krawędzi które znacząco odbiegają przepustowością od pozostałych (są wolniejsze)