

[Błądzenie losowe](#) [ang. [random walk](#)] to proces polegający na podejmowaniu kolejnych kroków, każdy w losowo wybranym kierunku.

W tym wypadku mieliśmy sprawdzić, czy prawdziwe jest twierdzenie, że liczba kroków agenta błądzącego losowo po nieskierowanym grafie spójnym po podstawieniu do odpowiedniego wzoru jest równa rezystancji zastępczej układu elektrycznego zbudowanego na podstawie grafu (z założeniem, że na każdej krawędzi kładziemy opornik o oporze  $1\Omega$ ). Wyniki są zaskakujące.

Prezentacja pokazuje wyniki testów jakie wykonaliśmy razem z Piotrkiem Grotą jako projekt na studia.

### Pliki do pobrania

[Pobierz pliki](#) (projekt Visual Studio, skrypt php).

Do wyliczania średnich używałem jeszcze jednego prostego skryptu PHP. Do wykresów wykorzystałem Open Office Calc.

### Krótki manual do programów

Program odpala się z wiersza poleceń. Na wejście podaje się:

{codecitation}liczba krawedzi

## Rundom Walk

Wpisany przez Patryk yarpo Jar  
poniedziałek, 15 lutego 2010 18:58

---

liczba wierzchołków

wierzchołek startowy

wierzchołek celu

liczba testów (później w programie zrobiłem jeszcze te liczby x 100)

v x - numery wierzchołów [numerowane od 0]

v u

...{/codecitation}

Czyli np.:

{codecitation}28

20

1

19

## Rundom Walk

Wpisany przez Patryk yarpo Jar  
poniedziałek, 15 lutego 2010 18:58

---

100

0 1

2 3

4 5

6 7

8 9

10 11

12 13

14 15

16 17

18 19

## Rundom Walk

Wpisany przez Patryk yarpo Jar  
poniedziałek, 15 lutego 2010 18:58

---

1 3

3 5

5 7

7 9

9 11

11 13

13 15

15 17

17 19

0 2

2 4

4 6

## Rundom Walk

Wpisany przez Patryk yarpo Jar  
poniedziałek, 15 lutego 2010 18:58

---

6 8

8 10

10 12

12 14

14 16

16 18{codecitation}

**Uwaga**: Bardzo ważny jest format. Między numerami wierzchołków musi być spacja. Kolejne krawędzie musi oddzielać enter, itp. Wejście musi być sformatowane jak powyższe. Nie chciało mi się bawić w scanfy w c# :) Wczytywanie danych jest bardzo uproszczone.