

System kolejkowy

Wykorzystywana literatura:

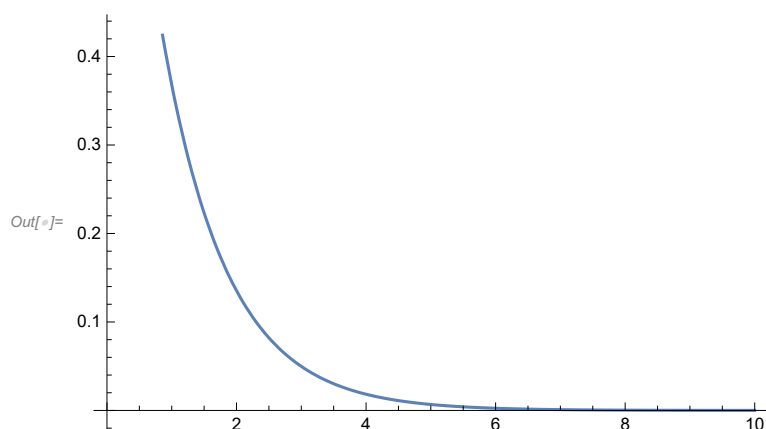
- Wykłady: <https://kacpertopol.github.io/>

Rozkład wykładniczy

FGP

$$f(x) = e^{-x\lambda} \lambda$$

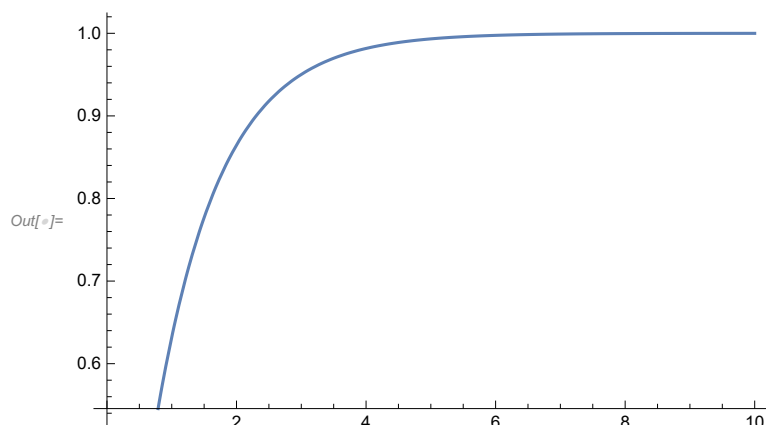
```
In[ ]:= λ = 1;  
Plot[λ * Exp[-x * λ], {x, 0, 10}]  
_wykres _funkcja eksponencjalna
```



Dystrybuanta CDF

$$F(x) = 1 - e^{-x\lambda}$$

```
In[ ]:= λ = 1;  
Plot[1 - Exp[-x * λ], {x, 0, 10}]  
_wykres _funkcja eksponencjalna
```



Odwracanie dystrybuanty

$$y = 1 - e^{-x\lambda}$$

$$1 - y = e^{-x\lambda}$$

$$\ln(y-1) = -x\lambda$$

$$-\frac{\ln(1-y)}{\lambda} = x$$

Odwrotna dystrybuanta

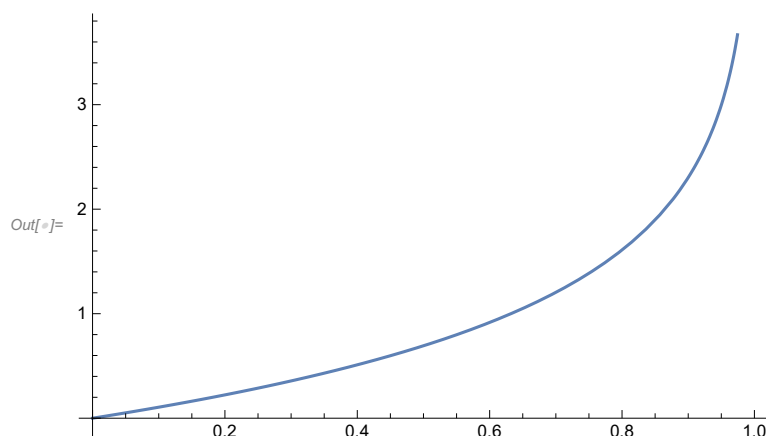
$$F^{-1}(x) = -\frac{\ln(1-y)}{\lambda},$$

gdzie $y \rightarrow \text{Uniform}(0,1)$

In[*]:= $\lambda = 1;$

Plot[$\frac{-\text{Log}[1-y]}{\lambda}$, {y, 0, 1}]

[wykres]



System kolejkowy

Tempo przychodzenia zadań do serwera: λ_A

Odstęp czasu pomiędzy przychodzeniem nowych zadań: $t_i^A = -\frac{\ln(1-n)}{\lambda_A}$, gdzie $n \rightarrow \text{Uniform}(0,1)$

Tempo wykonywania zadań przez serwer: λ_S

Czas wykonywania kolejnych zadań: $t_i^S = -\frac{\ln(1-n)}{\lambda_S}$, gdzie $n \rightarrow \text{Uniform}(0,1)$

Jednocześnie serwer może wykonywać tylko jedno zadanie.

Czas pojawiania się nowych zadań w systemie: $\alpha_i = t_1^A + t_2^A + \dots + t_i^A$

Czas gdy kolejne zadania będą wykonywane na serwerze: β_i

gdy serwer jest pusty: $\beta_i = \alpha_i$

gdy serwer zawiera kolejkę: $\beta_i = \beta_{i-1} + t_{i-1}^S$, gdzie $\beta_1 = t_1^A$

Przypadek $\lambda_A = \frac{1}{20}$ i $\lambda_S = \frac{1}{15}$ --- zadania wykonywane szybciej niż napływają

In[*]:= $\text{lambdaA} = 1/20 // N$

[przybliżenie n]

Out[*]:= 0.05

In[*]:= (*Średni czas czekania na kolejne zadanie:*)

In[*]:= Table[$-\frac{\text{Log}[1 - \text{RandomReal}[]]}{\lambda}$ /. $\lambda \rightarrow \text{lambdaA}$, {i, 1, 100000}] // Mean

[tabela]

[średnia arytmetyczna]

Out[*]:= 19.9981

```
In[ ]:= lambdaS = 1/15 // N
```

przybliżenie n

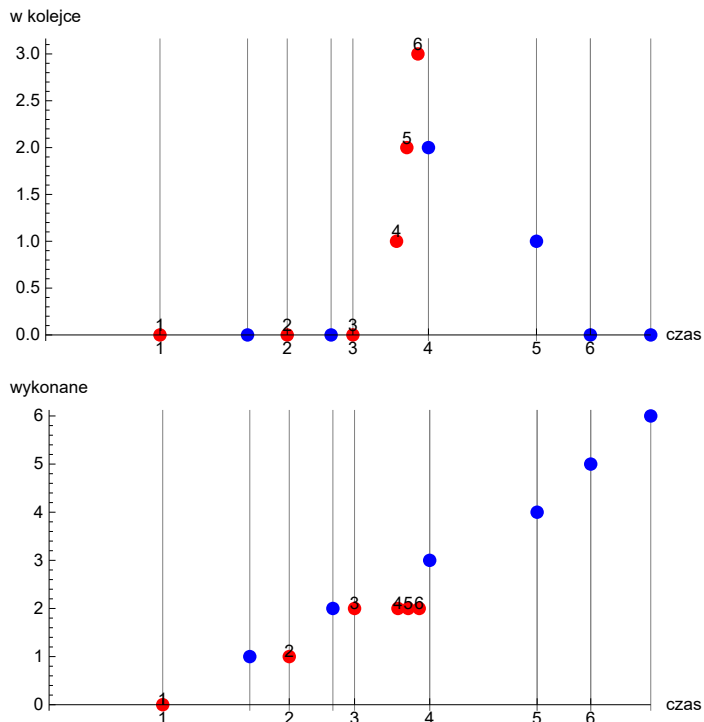
```
Out[ ]:= 0.0666667
```

```
In[ ]:= (*Średni czas czekania na wykonanie zadania:*)
```

```
In[ ]:= Table[-Log[1 - RandomReal[]] / . λ → lambdaS, {i, 1, 100000}] // Mean
```

tabela średnia arytmetyczna

```
Out[ ]:= 14.9938
```



Przypadek $\lambda_A = \frac{1}{20}$ i $\lambda_S = \frac{1}{100}$ --- zadania wykonywane wolniej niż napływają;
system się zatyka

```
In[ ]:= lambdaA = 1/20 // N
```

przybliżenie n

```
Out[ ]:= 0.05
```

```
In[ ]:= (*Średni czas czekania na kolejne zadanie:*)
```

```
In[ ]:= Table[-Log[1 - RandomReal[]] / . λ → lambdaA, {i, 1, 100000}] // Mean
```

tabela średnia arytmetyczna

```
Out[ ]:= 19.9324
```

```
In[ ]:= lambdaS = 1/100 // N
```

przybliżenie

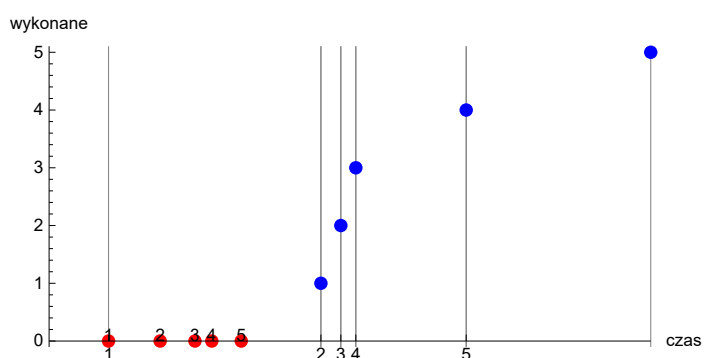
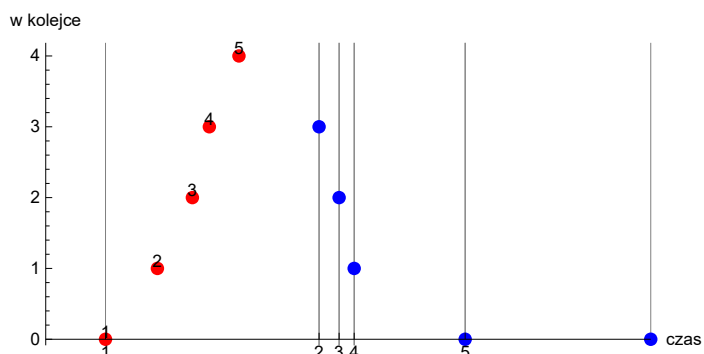
```
Out[ ]:= 0.01
```

```
In[ ]:= (*Średni czas czekania na wykonanie zadania:*)
```

```
In[*]:= Table[-  $\frac{\text{Log}[1 - \text{RandomReal[]}]}{\lambda}$  /.  $\lambda \rightarrow \text{lambdaS}$ , {i, 1, 100000}] // Mean
```

[tabela] [średnia arytmetyczna]

Out[*]:= 100.001



Przypadek $\lambda_A = \frac{1}{20}$ i $\lambda_S = \frac{1}{5}$ --- zadania wykonywane znacznie szybciej niż napływają

```
In[*]:= lambdaA = 1/20 // N
```

[przybliżenie n]

Out[*]:= 0.05

In[*]:= (*Średni czas czekania na kolejne zadanie:*)

```
In[*]:= Table[-  $\frac{\text{Log}[1 - \text{RandomReal[]}]}{\lambda}$  /.  $\lambda \rightarrow \text{lambdaA}$ , {i, 1, 100000}] // Mean
```

[tabela] [średnia arytmetyczna]

Out[*]:= 20.0451

```
In[*]:= lambdaS = 1/5 // N
```

[przybliżenie nui]

Out[*]:= 0.2

In[*]:= (*Średni czas czekania na wykonanie zadania:*)

```
In[*]:= Table[-  $\frac{\text{Log}[1 - \text{RandomReal[]}]}{\lambda}$  /.  $\lambda \rightarrow \text{lambdaS}$ , {i, 1, 100000}] // Mean
```

[tabela] [średnia arytmetyczna]

Out[*]:= 4.9944

