

## Sprawozdanie z problemu 5 filozofów

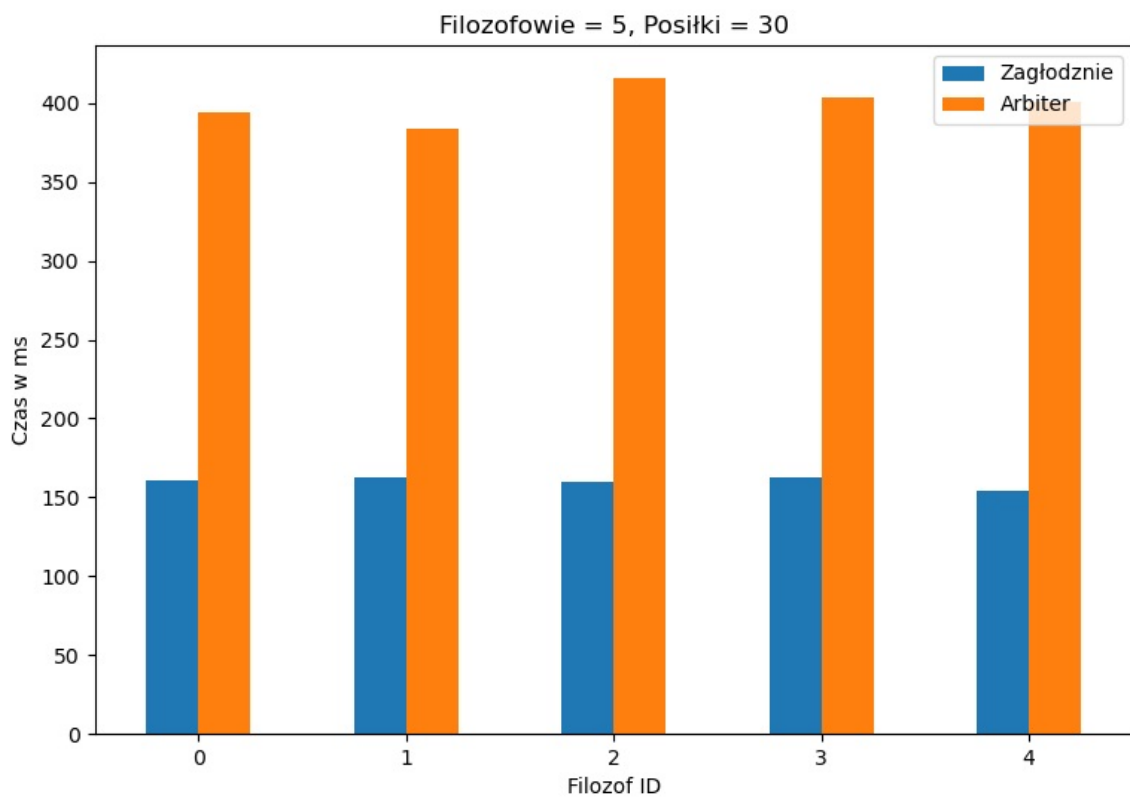
```
In [7]: import pandas as pd
        %matplotlib notebook
```

### Założenia:

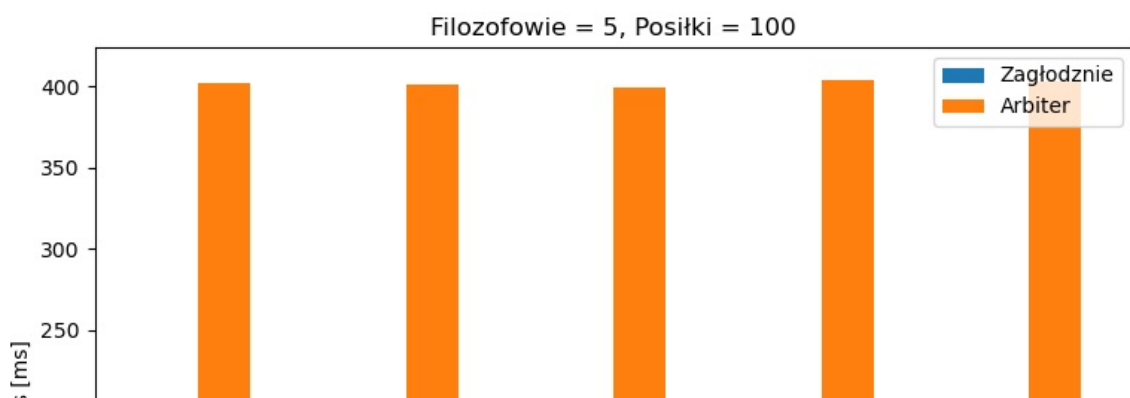
- czas jedzenia = 100ms
- czas odpoczynku = 0ms

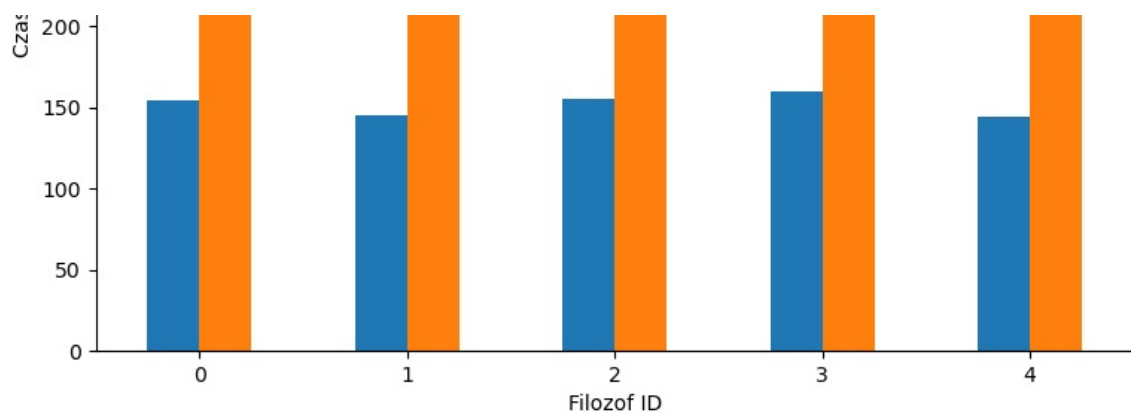
### Wykresy

```
In [55]: df_5_30 = pd.DataFrame({"Zagłodzenie": [161, 163, 160, 163, 154], "Arbiter": [394, 384, 416, 404, 401]})
df_5_30.plot.bar(figsize = (9,6), ylabel = "Czas w ms", xlabel = "Filozof ID", rot = 0,
                  title = "Filozofowie = 5, Posiłki = 30");
```

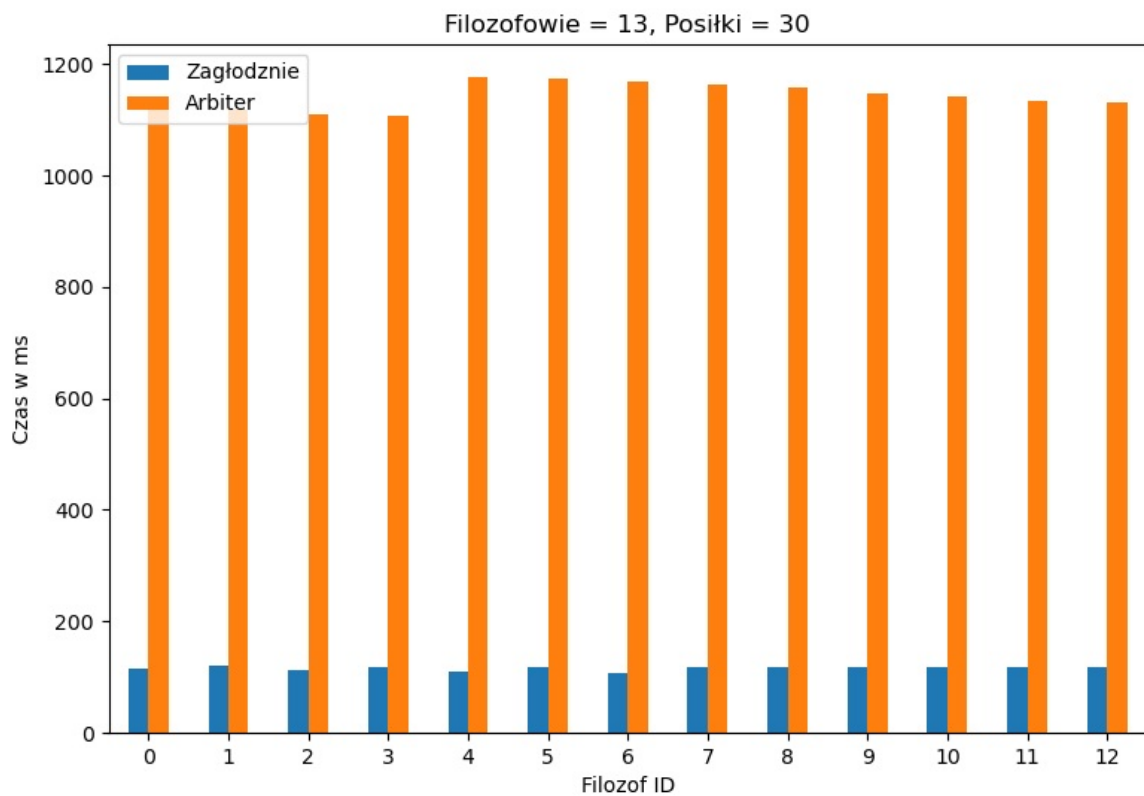


```
In [56]: df_5_100 = pd.DataFrame({"Zagłodzenie": [154, 145, 155, 160, 144], "Arbiter": [402, 401, 399, 404, 403]})
df_5_100.plot.bar(figsize = (9,6), ylabel = "Czas [ms]", xlabel = "Filozof ID", rot = 0,
                  title = "Filozofowie = 5, Posiłki = 100");
```



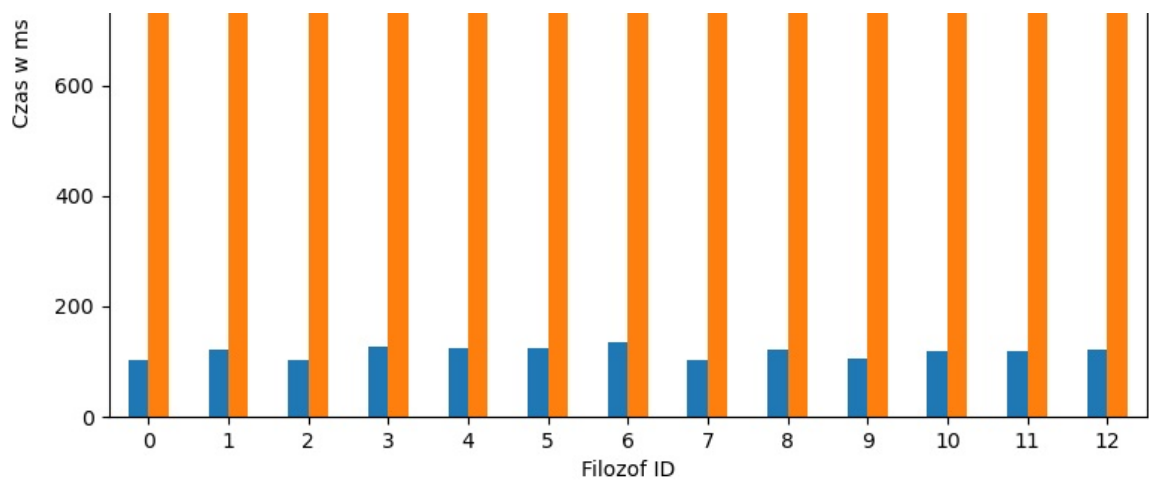


```
In [57]: df_13_30 = pd.DataFrame({"Zagłodzenie": [116, 121, 113, 119, 109, 119, 108, 117, 118, 117, 118, 117, 119],
                                "Arbiter":      [1121, 1118, 1111, 1108, 1177, 1175, 1169, 1164, 1158, 1148, 1141, 1134, 1131]})
df_13_30.plot.bar(figsize = (9,6), ylabel = "Czas w ms", xlabel = "Filozof ID", rot = 0,
                  title = "Filozofowie = 13, Posiłki = 30");
```



```
In [58]: df_13_100 = pd.DataFrame({"Zagłodzenie": [103, 122, 103, 126, 125, 124, 134, 102, 121, 106, 118, 118, 122],
                                    "Arbiter":      [1189, 1185, 1184, 1180, 1176, 1171, 1165, 1159, 1154, 1157, 1147, 1141, 1138]})
df_13_100.plot.bar(figsize = (9,6), ylabel = "Czas w ms", xlabel = "Filozof ID", rot = 0,
                  title = "Filozofowie = 13, Posiłki = 100");
```





### Podsumowanie

- Wykonałem tylko wersję z programowaniem wielowątkowym.
- Wersja z zagłodzeniem jest efektywniejsza niż wersja z arbitrem (za to w wersji z arbitrem zagłodzenie nie występuje).
- Im większa liczba filozofów tym dłuższy czas oczekiwania w wersji z arbitrem. Odwrotna zależność w wersji z zagłodzeniem, ale nie jest tak znacząca.