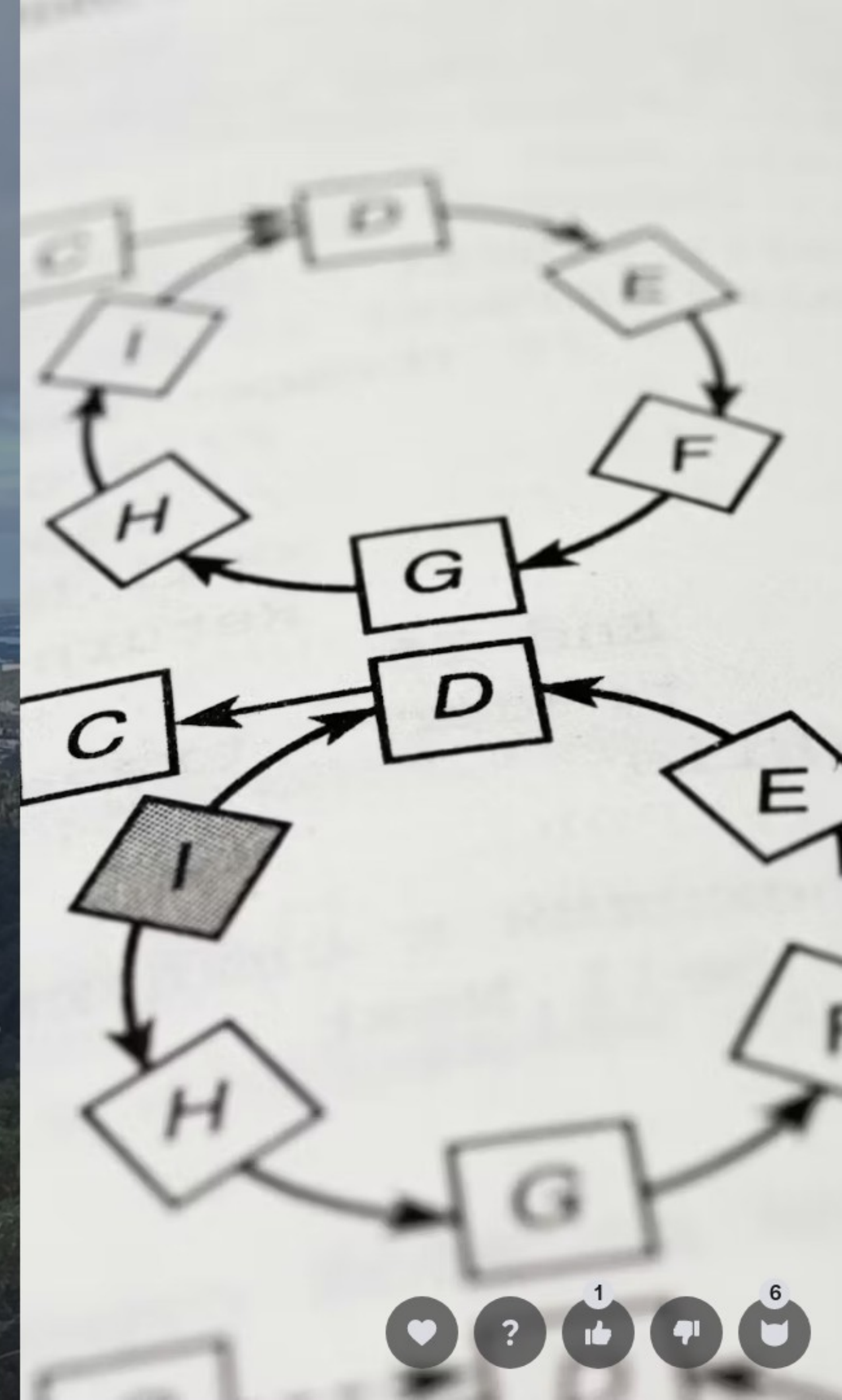
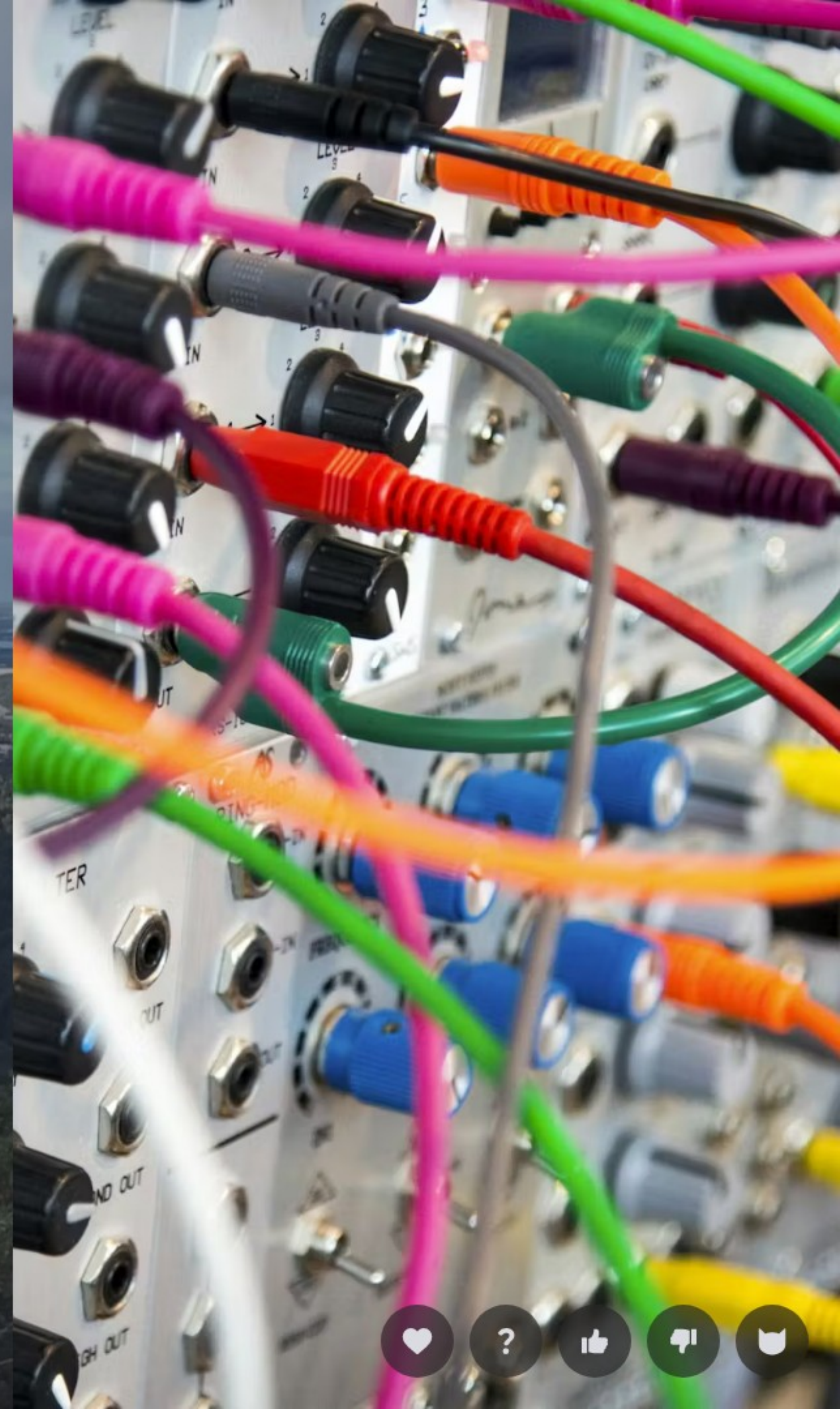


Algorytmy i Struktury Danych



Złożoność

czasowa i wiecznie zapominana pamięciowa



Jak porównać dwa algorytmy?

- Możemy mierzyć czas wykonania
- Możemy mierzyć dla różnych danych wejściowych
- Co, jeśli działają inaczej na różnych maszynach?
- Rozpatrzmy proste operacje, które wywołują się "natychmiast"
- Policzymy, ile takich operacji wykona każdy algorytm dla różnych danych
- To samo dla wielkości zajętej pamięci

Czym jest logarytm?

- Odwrotność potęgowania
- $\log_a(b) = c \iff a^c = b$
- Rośnie bardzo wolno
- \log_2 dla $n = 1, 1.000, 1.000.000, 1.000.000.000$ to $\approx 0, 10, 20$ i 30



Czy pojęcie logarytmu jest zrozumiałe?



TAK

0

NIE



4





Czym są O i Θ ?

- Jeśli dla wielkich n nasz algorytm nie będzie spowalniał bardziej niż $g(n)$, to jest on $O(g(n))$
- Jeśli wykonujemy pętle w pętli, obie n razy, to wykonamy n^2 operacji – działamy w $O(n^2)$
- Jeśli wykonujemy mniej operacji, to wciąż jesteśmy w $O(n^2)$
- Jeśli wykonujemy dokładnie n^2 operacji, to jesteśmy w $\Theta(n^2)$
- O to złożoność pesymistyczna – w najgorszym wypadku wykonamy tyle operacji (to nie jest poprawna definicja)
- Θ to złożoność dokładna – zawsze wykonamy tyle operacji
- Są jeszcze złożoności o (nadpesymistyczna), \sim (oczekiwana), Ω (optymistyczna) i ω (nadoptymistyczna)
- Wypróbujmy to dla przykładów

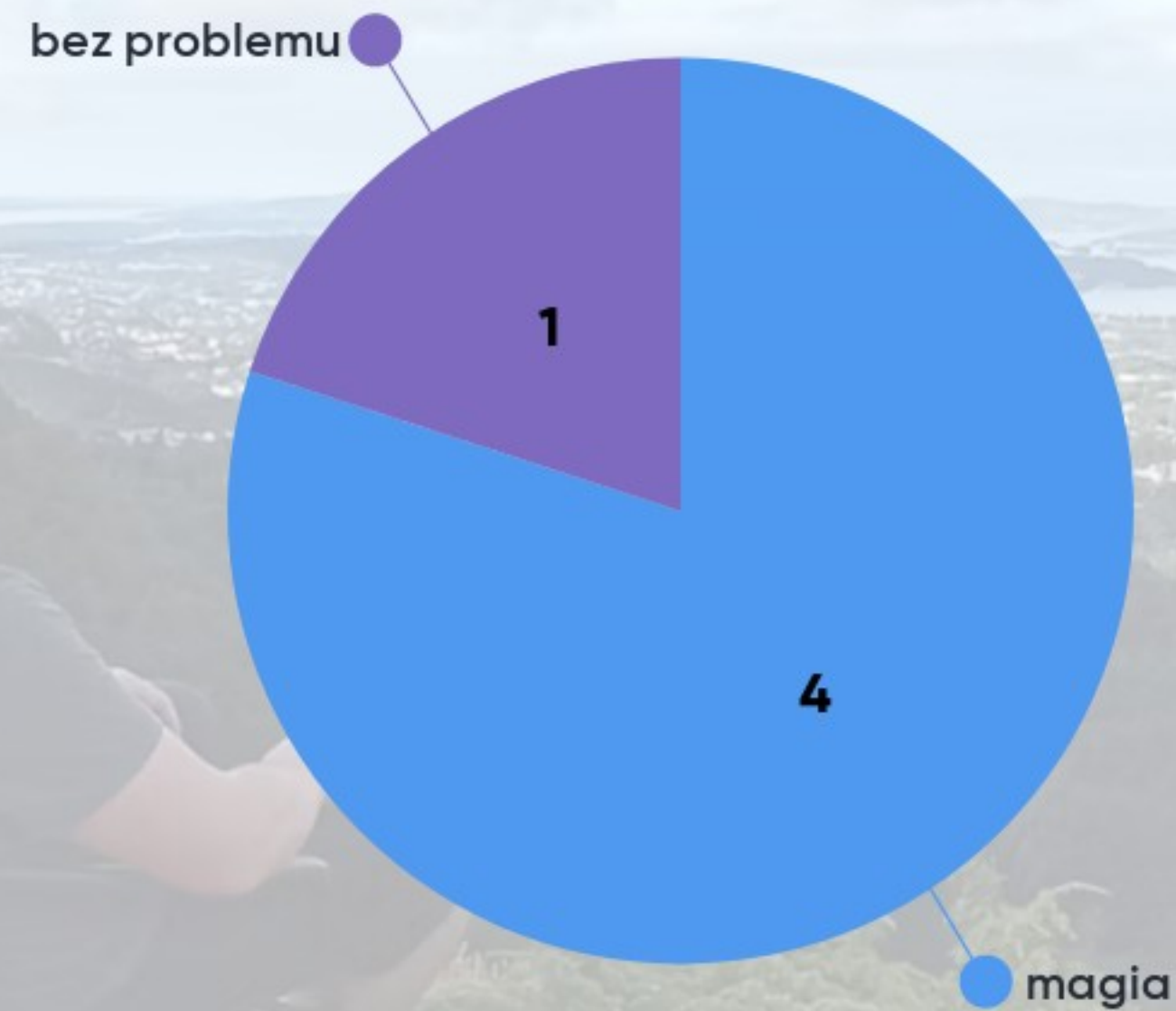
Jakie mamy złożoności?

- stała $\Theta(1) \approx O(1)$ – czas niezależny od n
- logarytmiczna $\Theta(\log n)$ – każdy krok pomniejsza n kilkukrotnie
- liniowa $\Theta(n)$ – wykonujemy tyle operacji, ile mamy elementów
- **!!! liniowo-logarytmiczna $\Theta(n \log n)$** – wykonujemy tyle operacji ile mamy elementów, powtarzając to dla pomniejszanych kilkukrotnie z każdym krokiem n
- kwadratowa $\Theta(n^2)$ – pętla w pętli, obie n -krotnie
- wielomianowa $\Theta(n^k)$ – kilka zagnieżdżonych pętli, mnożenie macierzy...
- wykładnicza $\Theta(c^n)$ – z każdym krokiem wykonujemy c razy więcej operacji
- silniowa $\Theta(n!)$ – sprawdzamy każdą możliwą kolejność n elementów

Złożoność obliczeniowa względem rozmiaru danych wejściowych

Rozmiar danych:	10	20	50	100	200	1000
$\log n$	3,32 ns	4,23 ns	5,64 ns	6,64 ns	7,64 ns	9,97 ns
n	10 ns	20 ns	50 ns	100 ns	200 ns	1 μ s
$n \log n$	33,21 ns	86,44 ns	282,2 ns	664,4 ns	1,53 μ s	9,97 μ s
n^2	100 ns	400 ns	2,5 μ s	10 μ s	40 μ s	1 ms
2^n	1 μ s	1,05 ms	13 dni	$4 \cdot 10^{13}$ lat	$5,1 \cdot 10^{43}$ lat	$3,4 \cdot 10^{284}$ lat
$n!$	3,6 ms	77 lat	$9,6 \cdot 10^{44}$ lat	$3 \cdot 10^{141}$ lat	$2,5 \cdot 10^{358}$ lat	$1,27 \cdot 10^{2551}$ lat

Jak się czujecie z tym tematem?



Algorytmy naiwne

