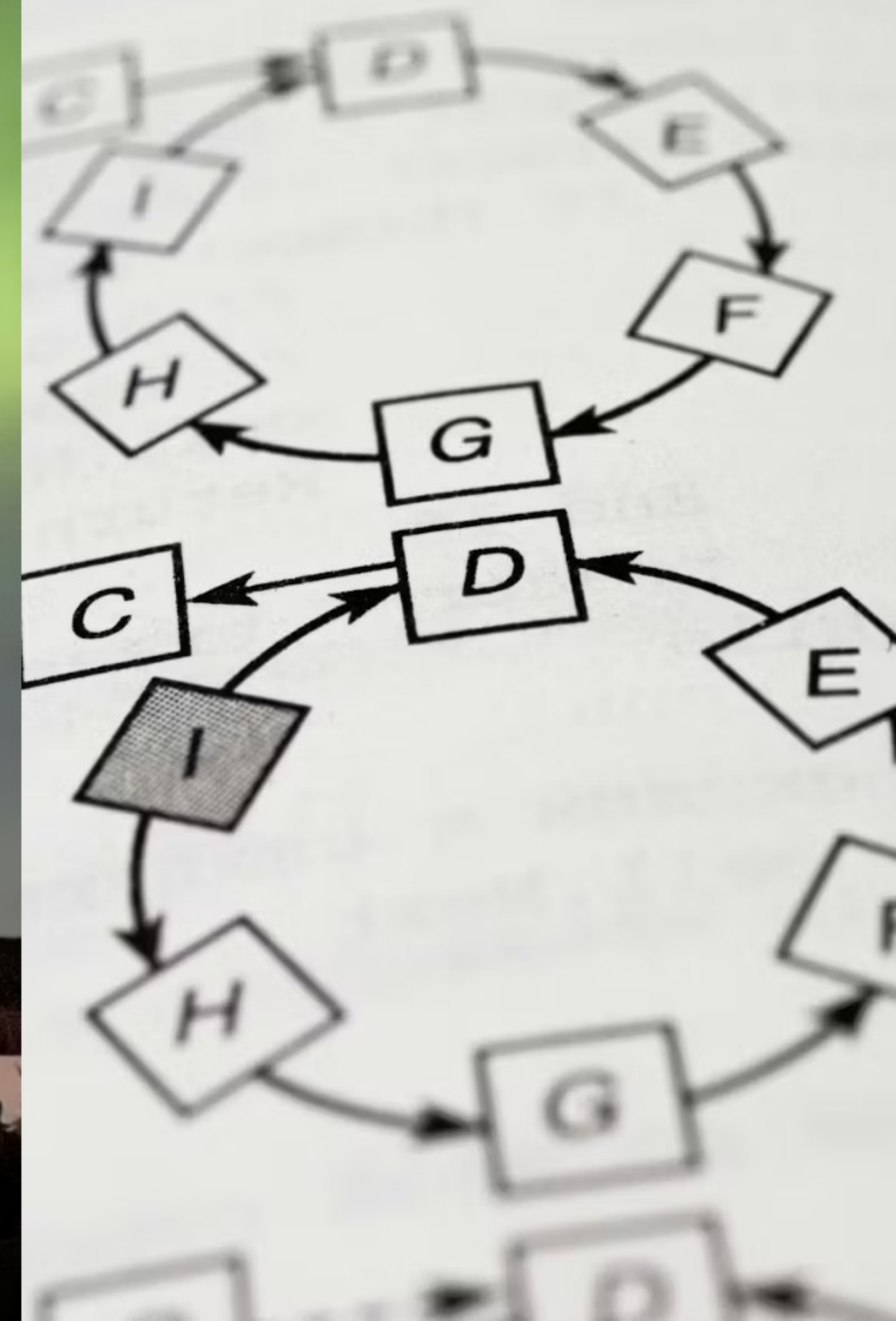


Algorytmy i Struktury Danych - wstęp



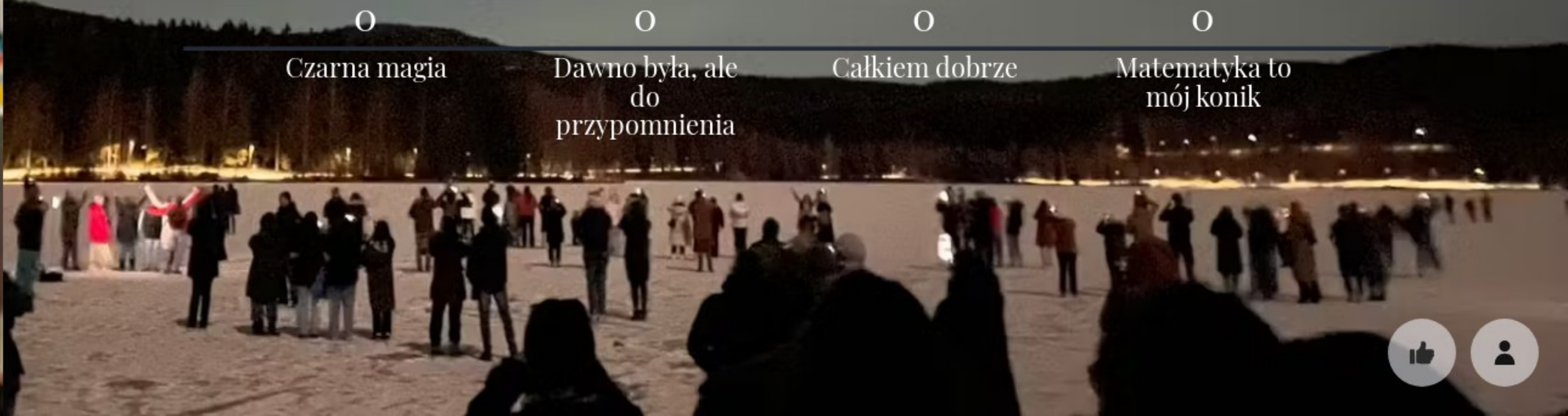
Kim jest prowadzący?

- Michał Jan Odorczyk
- Przygotowuję do konkursów algorytmicznych, olimpiad informatycznych i pomagam w projektach studenckich od 2015
- Java & TS developer w Capgemini (2019-2021)
- R developer w Lantern Machinery Analytics (2021-2022)
- Stażysta w dziale symulacji i wizualizacji w SINTEF Digital (2022)
- Pracuję dla SDA od 2022
- inż. Automatyki i Robotyki PWr
- lic. Informatyki UWr
- mgr Computational Science UiO
- (w trakcie) mgr Space Systems UiO
- Mam doświadczenie w C, C++, C#, F#, OCaml, Haskell, Assembly, Java, JavaScript, TypeScript, R, Python, Julia, Matlab...





Kim jesteście wy? Skąd przychodzicie? Czemu Python? Jak czujecie się z matematyką?



O
Czarna magia

O
Dawno była, ale
do
przypomnienia

O
Całkiem dobrze

O
Matematyka to
mój konik



Program


- Czym jest algorytm?
- Jak zapisać algorytm?
- Rekurencja
- Złożoność obliczeniowa i pamięciowa
- Rodzaje algorytmów
- Struktury danych
- Algorytmy powiązane ze strukturami danych





Czym jest algorytm?





Procedura rozwiązująca matematyczny problem w skończonym ciągu kroków, często uwzględniająca powtarzanie operacji. Szerzej: krok-po-kroku rozwiązywanie problemu lub osiąganie celu

– *Merriam-Webster Online Dictionary*



Algorytmicy przed komputerami

- Muḥammad ibn Mūsā al-Khwārizmī
- Ada Lovelace



Przykład

1. Niech indeks wskazuje na pierwszy element (początek) listy.
2. Niech x zawiera wartość elementu listy wskazywanego przez indeks (tzn. pierwszego).
3. Jeżeli zawartość elementu listy wskazywanego przez indeks jest większa od zawartości x , to przypisz x wartość elementu wskazywanego przez indeks.
4. Niech indeks wskazuje kolejny element listy; jeśli to niemożliwe (tzn. indeks wskazuje ostatni element listy, czyli jej koniec), przejdź do punktu 6.
5. Wróć do punktu 3.
6. Koniec.



Algorytm tworzenia algorytmu

1. Sformułowanie problemu/zadania
2. Ustalenie danych wejściowych
3. Określenie wyniku
4. Znalezienie metody wykonania zadania
5. Zapisanie algorytmu za pomocą wybranej metody
6. Analiza i testowanie algorytmu
7. Ocena skuteczności algorytmu



Jak zapisać algorytm?





Opis słowny

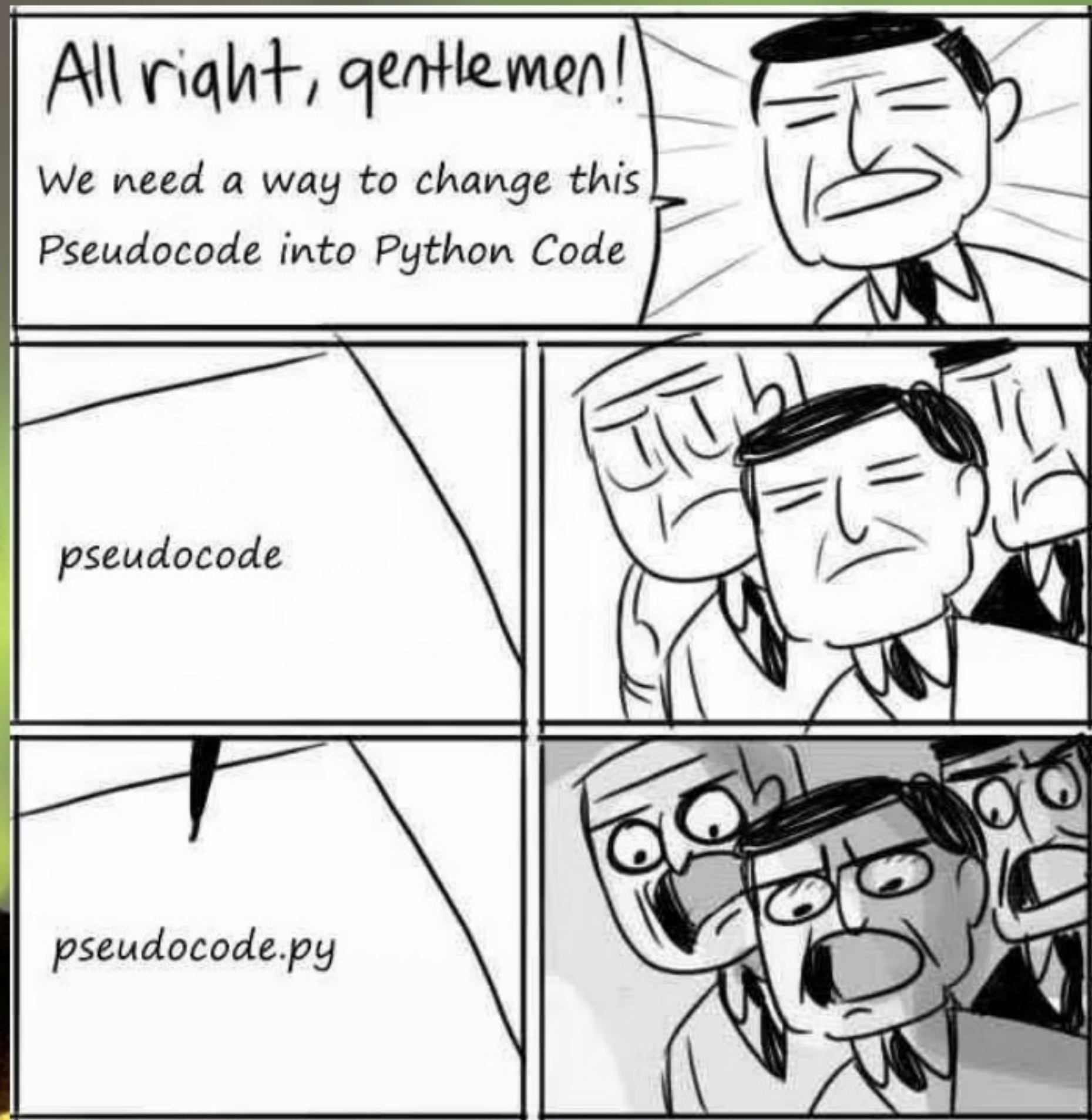
“Na początku zostaniesz umieszczony obok pierwszego sekretu. Aby do niego dotrzeć, odwróć się i na lewej ścianie znajdziesz rozbity kamień, którego mniejsza część to przycisk. Wciśnij go, by otworzyć ścianę, za którą znajduje się shuriken i cichoczłapy. Oba mogą być bardzo przydatne dla łotrzyka w twojej drużynie. Jeśli ciemność utrudnia odnalezienie ukrytego przycisku, zacznij od udania się do punktu (2).”



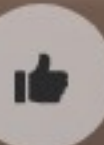
1. Mleko (lub wodę), kakao i cukier przełożyć do garnka i mieszając, zagotować. Do gorącej masy dodać masło i mieszać, aż się rozpuści. Pozostawić do ostygnięcia.
2. Mąkę wymieszać z proszkiem do pieczenia. Odstawić na bok.
3. Jajka sparzyć wrzątkiem. Oddzielić żółtka od białek. Białka ubić na sztywną pianę. Odstawić na bok. Żółtka wmieszać do ostygniętej masy, trzepaczką lub mikserem. Odląć pół szklanki masy. Ostawić na bok. (Będzie to polewa). Trzepaczką (lub mikserem) wmieszać mąkę z proszkiem. Na końcu wmieszać delikatnie szpatułką pianę z białek.
4. Dno tortownicy o średnicy 26cm wyłożyć papierem do pieczenia, a następnie zacisnąć obręcz. Ciasto przełożyć do formy.
5. Piec w nagrzanym piekarniku ok. 45 minut, do suchego patyczka, w temperaturze 180°C, grzałka góra- dół. Pozostawić do ostygnięcia.
6. Ciasto polać odłożoną polewą. (Gdyby polewa była za rzadka, należy włożyć ją na chwilę do lodówki, aż lekko zgęstnieje).

Lista kroków





Pseudokod



Algorithm Ford-Fulkerson

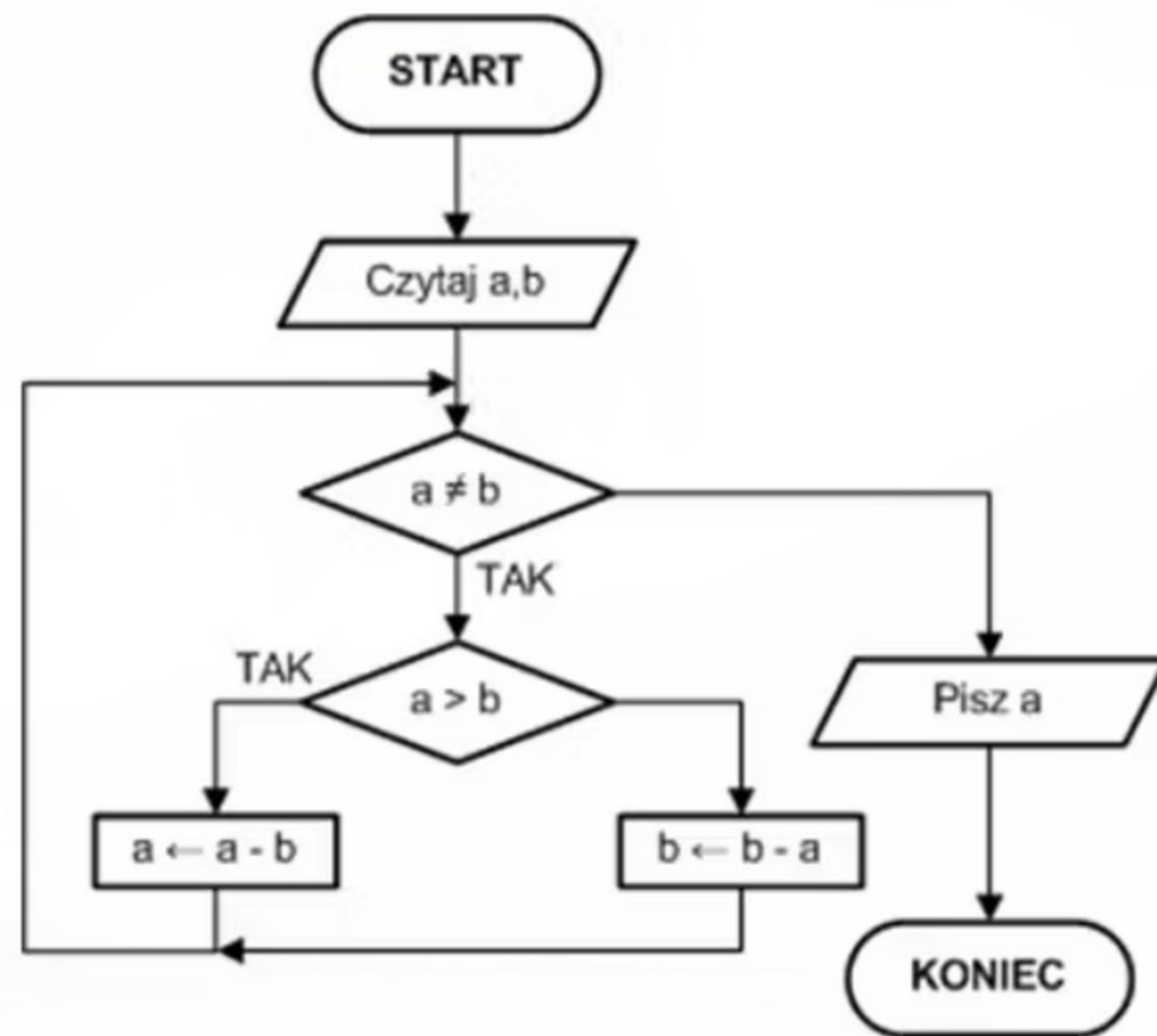
Inputs Given a Network $G = (V, E)$ with flow capacity c , a source node s , and a sink node t

Output Compute a flow f from s to t of maximum value

1. $f(u, v) \leftarrow 0$ for all edges (u, v)
2. While there is a path p from s to t in G_f , such that $c_f(u, v) > 0$ for all edges $(u, v) \in p$:
 1. Find $c_f(p) = \min\{c_f(u, v) : (u, v) \in p\}$
 2. For each edge $(u, v) \in p$
 1. $f(u, v) \leftarrow f(u, v) + c_f(p)$ (*Send flow along the path*)
 2. $f(v, u) \leftarrow f(v, u) - c_f(p)$ (*The flow might be "returned" later*)

- " \leftarrow " denotes **assignment**. For instance, " $largest \leftarrow item$ " means that the value of *largest* changes to the value of *item*.
- "**return**" terminates the algorithm and outputs the following value.

Pseudokod



Schematy blokowe



Rekurencja

