Programowanie I

* algorytmy i struktury danych
* złożoność obliczeniowa
* sortowanie

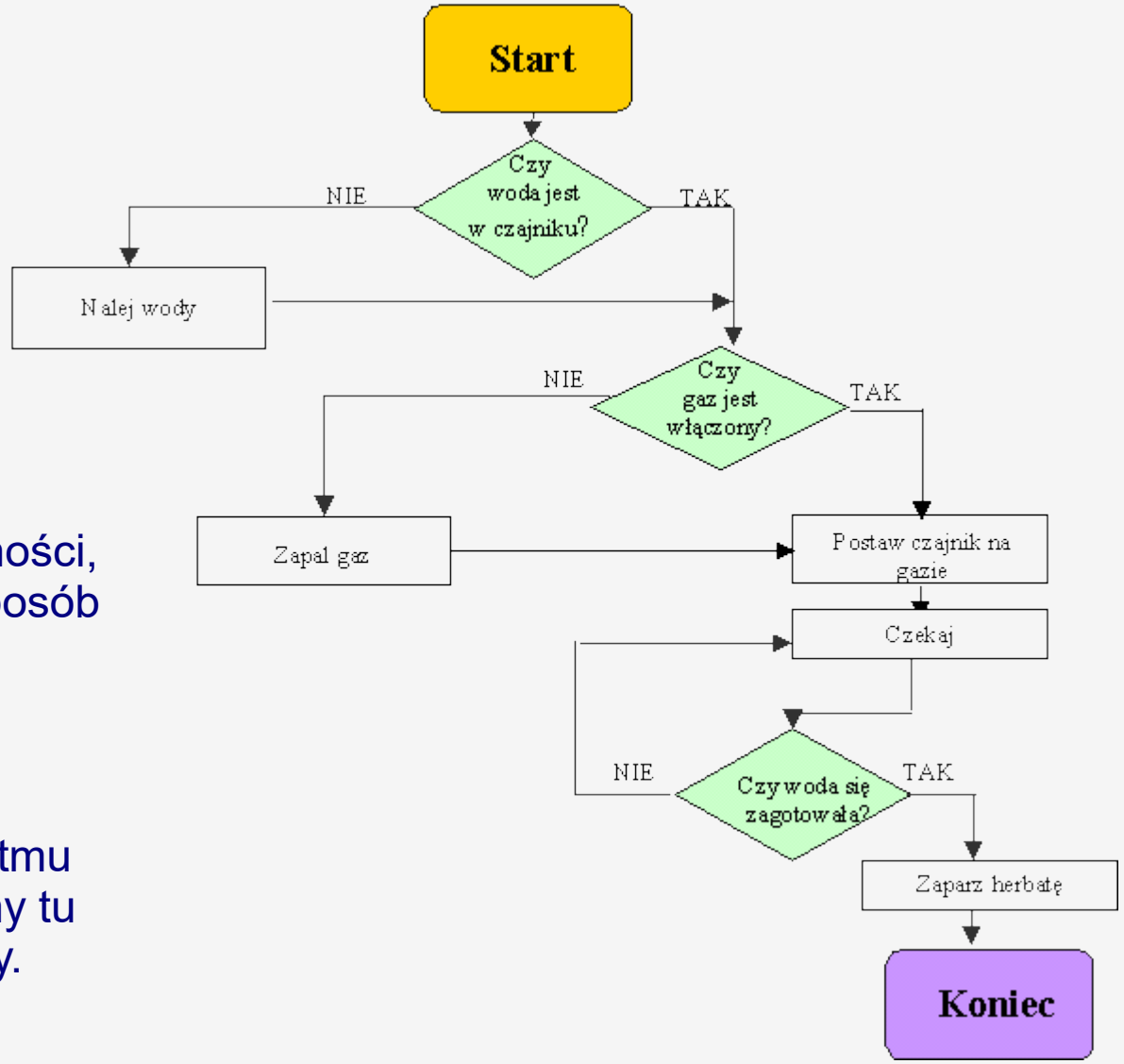
Mateusz „Divo” Górny
szkolenia@divo.net.pl

Algorytm – co to?

- Algorytm :

Osiągnięcie danego celu poprzez wykonanie krok po kroku każdego działania

Algorytm - skończony ciąg jasno zdefiniowanych czynności, koniecznych do wykonania pewnego rodzaju zadań. Sposób postępowania prowadzący do rozwiązania problemu. Zadaniem algorytmu jest przeprowadzenie systemu z pewnego stanu początkowego do pożądanego stanu końcowego.
Jako przykład stosowanego w życiu codziennym algorytmu podaje się często przepis kulinarny. Dla przykładu mamy tu przepis na zagotowanie wody do przygotowania herbaty.



Algorytmika

- Algorytmika:

Nauka o algorytmach.

Dział informatyki, cybernetyki, matematyki, nauk przyrodniczych i wielu innych.

Algorytmika jest nauką o algorytmach. Patrząc na ogrom dziedzin nauki, w których występuje można stwierdzić, że algorytmy są potrzebne praktycznie wszędzie.
Algorytm genetyczny – jest w stanie się sam rozwijać, ucząc się na własnych błędach.
Algorytmy sztucznej inteligencji – są w stanie uczyć się jak najlepszego rozwiązania danego problemu
Algorytmy wykorzystujące uczenie maszynowe – uczą się wyciągać wnioski (np. wykorzystanie w mapach google analizując sytuacje na drogach)

- Przykłady algorytmów:

* genetyczne

* sztucznej inteligencji

* wykorzystujące uczenie maszynowe

Algorytm – dlaczego tak ważny?

- Po co?

  * Ułatwia pracę
  * Pomaga zrozumieć sposób „myślenia” komputera
  * Uczy tworzyć perfekcyjny kod

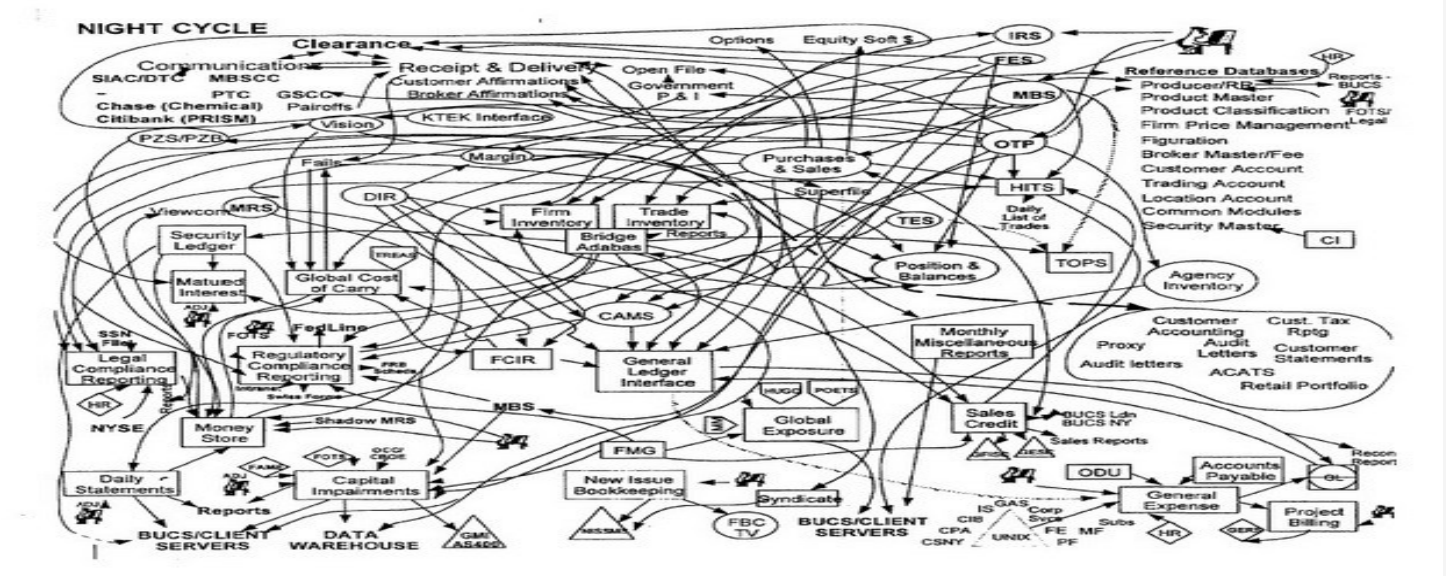
Dojść do perfekcji:

  * Wyznaczenie celu
  * Opracowanie skryptu
  * Utworzenie kodu poszczególnych kroków

- Taki algorytm

  nam nie pomoże ;)

Stworzenie algorytmu pomoże nam krok po kroku opracować dobry kod. Początkujący programiści chcą czasami szybko zacząć pisać. To jest bardzo duży błąd. W momencie kiedy zaczniemy najpierw tworzyć algorytm, będziemy mogli podzielić sobie pracę na odpowiednie etapy. Zadanie wtedy wydaje się bardziej przejrzyste i czasami łatwiejsze. W dodatku będziemy dokładnie wiedzieli, czego potrzebuje nasz program, nie ominiemy ważnych elementów i co najważniejsze – nauczymy się myśleć jak komputer.



Alan Turing – kto to?

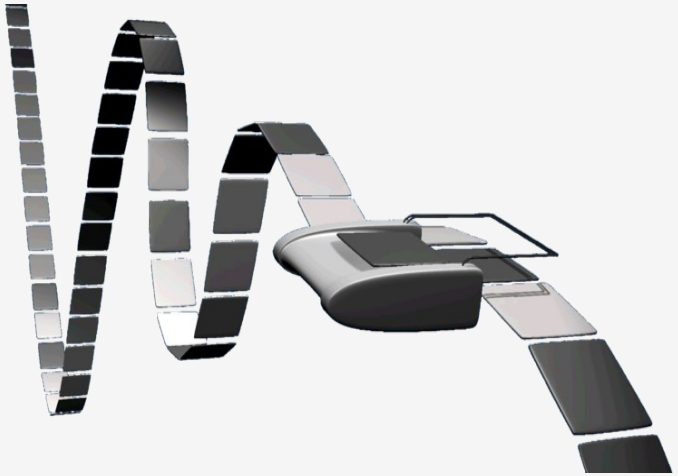
* angielski matematyk i kryptolog
* twórca bomby Turinga
* ojciec sztucznej inteligencji
* twórca maszyny Turinga

Osobą ważną dla agorytmiki jest:
Alan Turing - angielski matematyk, kryptolog, twórca pojęcia maszyny Turinga i jeden z twórców informatyki. Uważany za ojca sztucznej inteligencji.
Twórca bomby Turinga (urządzenia służącego do łamania kodu Enigmy).

Stworzył również Maszynę Turinga abstrakcyjny model komputera służącego do wykonywania algorytmów, składającego się z nieskończenie długiej taśmy podzielonej na pola w których zapisuje się dane. Taśma może być nieskończona jednostronnie lub obustronnie. Każde pole może znajdować się w jednym z N stanów. Maszyna zawsze jest ustawiona nad jednym z pól i znajduje się w jednym z M stanów. Zależnie od kombinacji stanu maszyny i pola maszyna zapisuje nową wartość w polu, zmienia stan, a następnie może przesunąć się o jedno pole w prawo lub w lewo. Taka operacja nazywana jest rozkazem. Maszyna Turinga jest sterowana listą zawierającą dowolną liczbę takich rozkazów. Liczby N i M mogą być dowolne, byle skończone. Czasem dopuszcza się też stan M+1, który oznacza zakończenie pracy maszyny. Lista rozkazów dla maszyny Turinga może być traktowana jako jej program.

Maszyna Turinga

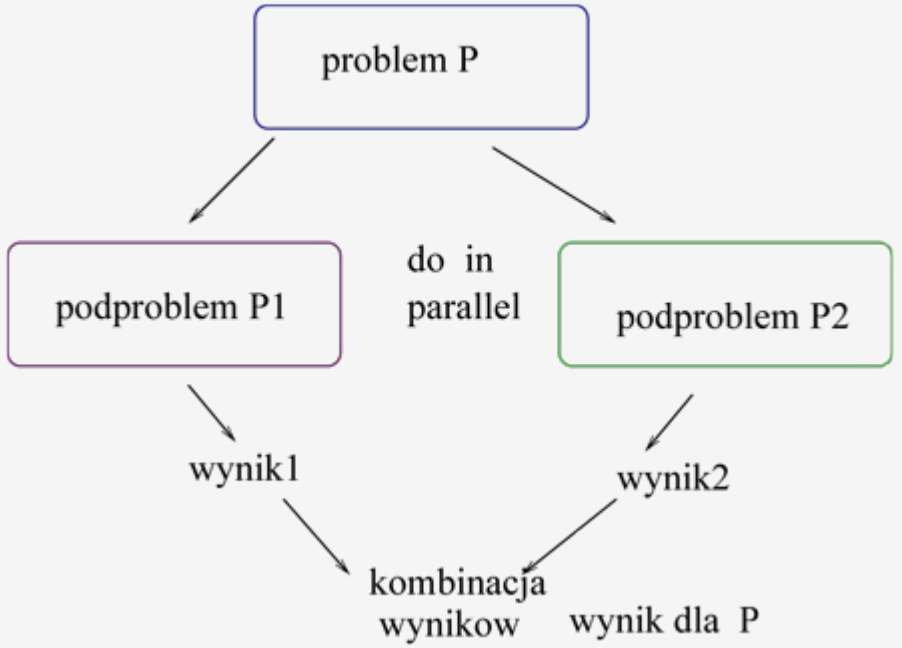
- Nieskończenie długa taśma podzielona na pola, na których jest zapis danych
- Każde pole znajduje się w jednym z N stanów
- Maszyna zawsze jest ustawiona na jednym z pół i znajduje się w jednym z M stanów
- Artystyczna wizji maszyny Turinga



Klasyfikacja algorytmów

- * Dziel i zwyciężaj
- * Programowanie dynamiczne
- * Metoda zachłanne
- * Programowanie liniowe
- * Wyszukiwanie wyczerpujące
- * Heurystyka

- Przykład algorytmu dziel i zwyciężaj:



Uzupełnienie slajdu wyżej:

Podstawowe paradygmaty tworzenia algorytmów komputerowych:
dziel i zwyciężaj – dzielimy problem na kilka mniejszych, a te znowu dzielimy, aż ich rozwiązania staną się oczywiste;
programowanie dynamiczne – problem dzielony jest na kilka, ważność każdego z nich jest oceniana i po pewnym wnioskowaniu wyniki analizy niektórych prostszych zagadnień wykorzystuje się do rozwiązania głównego problemu;
metoda zachłanna – nie analizujemy podproblemów dokładnie, tylko wybieramy najbardziej obiecującą w danym momencie drogę rozwiązania;
programowanie liniowe – oceniamy rozwiązanie problemu przez pewną funkcję jakości i szukamy jej minimum;
wyszukiwanie wyczerpujące – przeszukujemy zbiór danych, aż do odnalezienia rozwiązania;
heurystyka – człowiek na podstawie swojego doświadczenia tworzy algorytm, który działa w najbardziej prawdopodobnych warunkach, rozwiązanie zawsze jest przybliżone.

Techniki implementacji algorytmów

* Proceduralność
* Praca równoległa
* Rekurencja
* Obiektowość
* Algorytm probabilistyczny

Najważniejsze techniki implementacji algorytmów komputerowych:
proceduralność – algorytm dzielimy na szereg podstawowych procedur, wiele algorytmów współdzieli wspólne biblioteki standardowych procedur, z których są one wywoływane w razie potrzeby;
praca równoległa – wiele procedur wykonywanych jest w tym samym czasie, wymieniają się one danymi;
rekurencja – procedura lub funkcja wywołuje sama siebie, aż do uzyskania wyniku lub błędu;
obiektowość – procedury i dane łączymy w pewne klasy reprezentujące najważniejsze elementy algorytmu oraz stan wewnętrzny wykonującego je systemu;
algorytm probabilistyczny – działa poprawnie z bardzo wysokim prawdopodobieństwem, ale wynik nie jest pewny