

1

Jak do tego wszystkiego
doszło?

Od obliczeń do
komunikacji

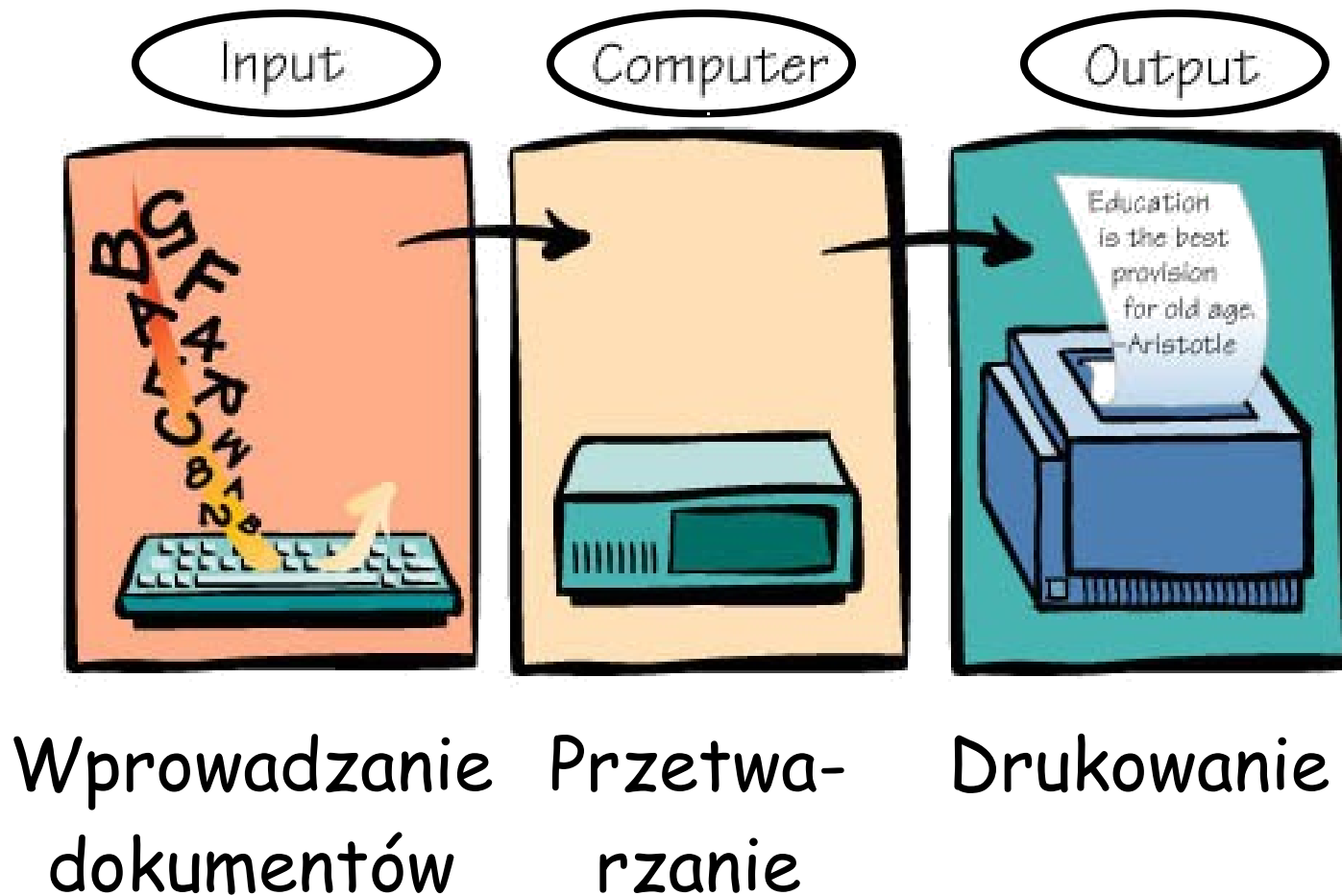
Życie bez komputerów?



Komputery
ingerują w nasze
życie prawie
wszędzie



Typowy przepływ informacji

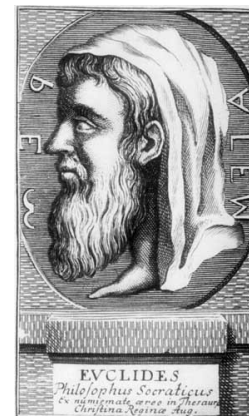


Pierwsi informatycy

- W sposób uświadomiony metody algorytmiczne stosowali starożytni Grecy
- Opracowali całą teorię konstrukcji geometrycznych o całym praktycznych zastosowaniach

Pierwsi twórcy algorytmów

- Euklides (~365-~300pne): algorytm obliczania największego wspólnego dzielnika
- Eratosthenes (~275-~194pne): algorytm wyznaczania liczb pierwszych - sito Eratostenesa



Wyznaczanie liczb pierwszych

- Liczba pierwsza, to taka liczba, która ma dokładnie dwa różne podzielniki: jedność i samą siebie, np 2,3,5,7,11,...`
- Wcale nie jest łatwo:
 - stwierdzić, czy liczba jest pierwsza
 - wyznaczyć kolejną liczbę pierwszą
 - wiedząc, że liczba nie jest pierwsza znaleźć jej dzielniki

Sito Eratostenesa

- Wypisz odpowiednio dużo liczb naturalnych poczynając od 2
- Wykreśl wszystkie liczby podzielne przez 2
- Kolejno pobieraj pierwsze niewykreślone liczby i wykreślaj ich wielokrotności.

Przykład – wypisujemy odpowiednio dużo kolejnych liczb

- 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39 40 41
42 43 44 45 46 47 48 49 50 51 52 53
...

Przykład – wykreślamy wielokrotności 2

- 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39 40 41
42 43 44 45 46 47 48 49 50 51 52 53
...

Przykład - wykreślamy wielokrotności 3

- 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39 40 41
42 43 44 45 46 47 48 49 50 51 52 53
...

Przykład – wykreślamy wielokrotności 5

- 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39 40 41
42 43 44 45 46 47 48 49 50 51 52 53
...

Przykład - wykreślamy wielokrotności 7

- 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39 40 41
42 43 44 45 46 47 48 49 50 51 52 53

...

Zostały już tylko liczby pierwsze

Problemy z algorytmem Eratostenesa

- Nadaje się tylko do znajdowania stosunkowo niewielkich liczb
- Wymaga dodatkowej pamięci na przechowywanie kolejnych liczb naturalnych - jest **złożony pamięciowo**.

Największa liczba pierwsza do dziś odkryta

- $2^{43,112,609}-1$ jest największą do 13.10.2008 znaną liczbą pierwszą. Została ona odkryta w 08.08.2008 przez GIMPS. Składa się ona z ok. 12.9 miliona cyfr.
- Użyta metoda jest zupełnie inna, niż za pomocą sita Eratostenesa.

Problem wyznaczenia największego wspólnego dzielnika

- Dane są dwie liczby naturalne m, n
- Znajdź największą liczbę naturalną, przez którą dzieli się bez reszty zarówno m jak i n .
- Na przykład dla $m=18$ i $n=24$
 $NWD(m, n)=6$

Algorytm Euklidesa (1)

- $0 \leq m \leq n, \quad n > 0$
- Jeśli $m=0$ to $NWD(m,n)=n$
- Jeśli $m>0$ to $NWD(m,n)=NWD(m,n-m)$

Wada algorytmu Euklidesa (1)

- Jest bardzo wolny - przy złośliwych danych, czyli wtedy, gdy jedna z liczb będzie bardzo duża, a druga będzie jedynką, trzeba odejmować od większej z nich jedynkę tyle razy, ile wynosi wartość większej liczby.
- ... co już dla liczb 30-cyfrowych jest zupełnie niewykonalne!

Algorytm Euklidesa (2)

- $0 \leq m \leq n, n > 0$
- Jeśli $m=0$ to $\text{NWD}(m,n)=n$
- Jeśli $m>0$ to

$$\text{NWD}(m,n) = \text{NWD}(n \bmod m, m)$$

Gdzie $n \bmod m$ to reszta z dzielenia n przez m

Wady i zalety

- Wada: programuje się trudniej (dzielenie trudniej zaprogramować niż odejmowanie)
- Zaleta: wykonuje się szybko - proporcjonalnie do długości zapisu liczby w systemie dziesiętnym.

Algorytm Euklidesa (3)

- Jeśli $m=0$ to $NWD(n,m) = n$
- Jeśli $n=0$ to $NWD(n,m)=m$
- Jeśli $n,m \in P$ to $NWD(n,m)=2NWD(n/2,m/2)$
- Jeśli n jest parzysta, a m nieparzysta, to $NWD(n,m)=NWD(n/2,m)$
- Jeśli n jest nieparzysta, a m parzysta, to $NWD(n,m)=NWD(n,m/2)$
- Jeśli n,m są nieparzyste, to $NWD(n,m)=NWD(n-m,m)$ dla $n \geq m$ lub $NWD(n,m)=NWD(m-n,n)$ dla $m \geq n$

Wady i zalety

- Wady:
 - ma skomplikowany opis
 - wykonuje się nieco dłużej niż poprzedni algorytm
- Zaleta: nadal wykonuje się szybko; proporcjonalnie do długości zapisu liczb.

Podsumowanie algorytmów Euklidesa

- Algorytm (1) jest bezużyteczny w praktyce, chyba że ktoś ma do czynienia z małymi liczbami. Jego złożoność czasowa jest tak duża, że dla danych spotykanych np. w kryptografii nie byłibyśmy w stanie wykonać podstawowych kroków algorytmów szyfrujących.

Podsumowanie algorytmów Euklidesa -cd

- Algorytmy (2) i (3) są porównywalne, jeśli chodzi o jakość; częściej stosuje się algorytm (2).
- Pozwalają one swobodnie obliczać NWD nawet dla liczb kilkusetcyfrowych w rozsądnym czasie.

Dziedziny algorytmiczne

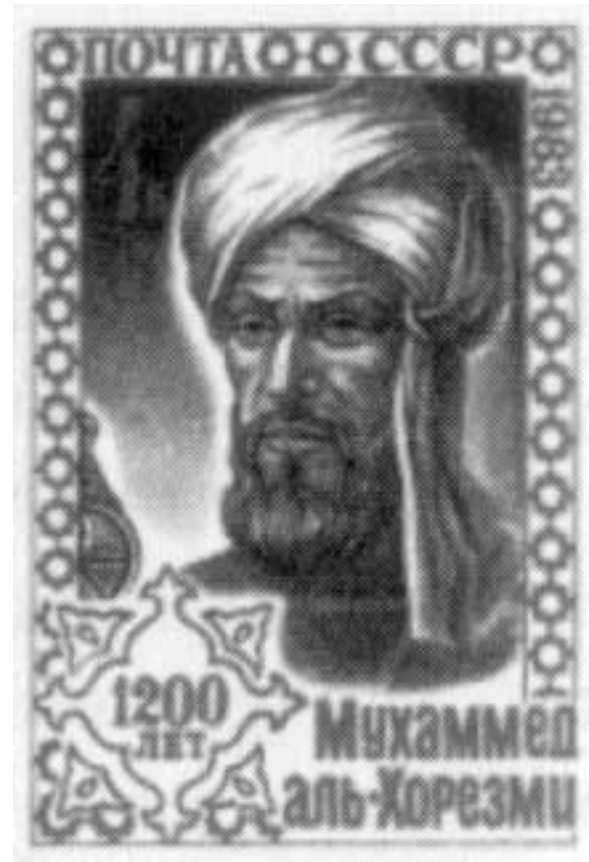
- Euklides 1: $(N, =0, \leq, -)$
- Euklides 2: $(N, =0, \text{mod})$
- Euklides 3: $(N, =0, \leq, \text{Par?}, *2, /2, -)$

Przykład

- $\text{NWD}(36,84) = \text{NWD}(36,48) =$
 $= \text{NWD}(36,12) = \text{NWD}(24,12) =$
 $= \text{NWD}(12,12) = \text{NWD}(12,0) = 12$
- $\text{NWD}(36,84) = \text{NWD}(12,36) = \text{NWD}(0,12) = 12$
- $\text{NWD}(36,84) = 2 * \text{NWD}(18,42) = 4 * \text{NWD}(9,21) =$
 $4 * \text{NWD}(12,21) = 4 * \text{NWD}(6,21) = 4 * \text{NWD}(3,21) =$
 $4 * \text{NWD}(3,18) = 4 * \text{NWD}(3,9) =$
 $4 * \text{NWD}(3,6) = 4 * \text{NWD}(3,3) = 4 * \text{NWD}(0,3) = 4 * 3$
 $= 12$

Al Chwarizmi

- Abu Ja'far
Muhammad ibn
Musa Al-
Chwarizmi (~780-
~850)
- *Hisab al-jabr
w'al-muqabala*
- *Opisał ciekawe
algorytmy*



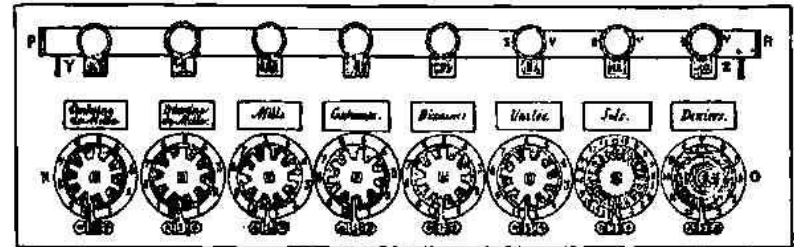
Wilhelm Schickard 1592-1635

- Zbudował pierwszy kalkulator



Blaise Pascal 1623-1662

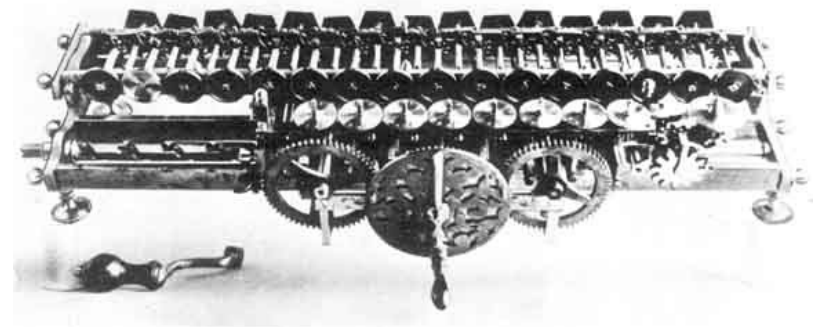
- Wybitny matematyk, fizyk, teolog, filozof, konstruktor
- Między innymi zajmował się konstrukcją kalkulatorów



Gotfried Wilhelm Leibniz

1646-1716

- Wybitny filozof, matematyk, prawnik i konstruktor
- Udoskonalił kalkulator
- Wymyślił mechanizm zwany kołem Leibniza

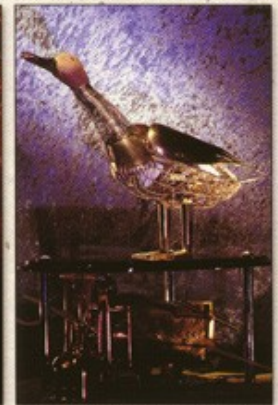


Jacques de Vaucanson (1709-82)

- Wielki francuski konstruktor i wynalazca. Ojciec robotyki
- Skonstruował wiele mechanicznych zabawek
- Jako pierwszy w historii użył kart perforowanych do zapisu danych



MUSEE des AUTOMATES de GRENOBLE



Hommage à VAUCANSON

Joseph Jacquard (1752-1834)



- twórca krosna tkackiego,
- wykorzystał pomysły de Vaucansona
- używał kart perforowanych do kodowania wzorów tkackich



Karty perforowane Jacquarda

**Z tego zestawu
dziurek powstaje
piękna tkanina**



Farkas von Kempelen (1734-1804)

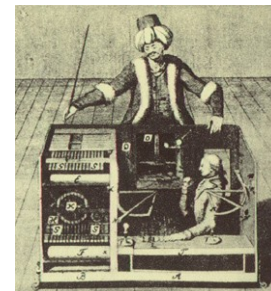


Wynalazca węgierski.
Twórca automatu szachowego
MEPHISTO

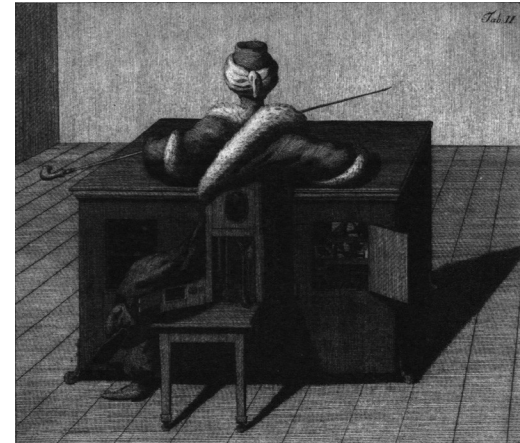
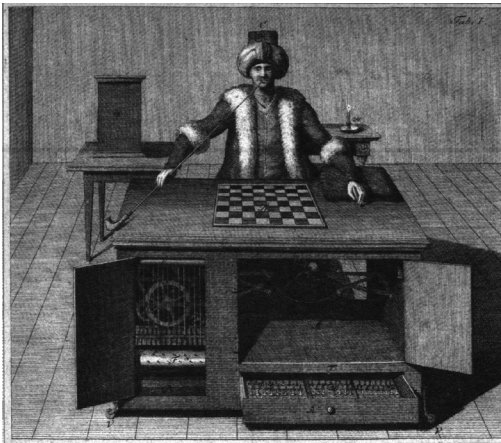


Farkas von Kempelen 1734-1804

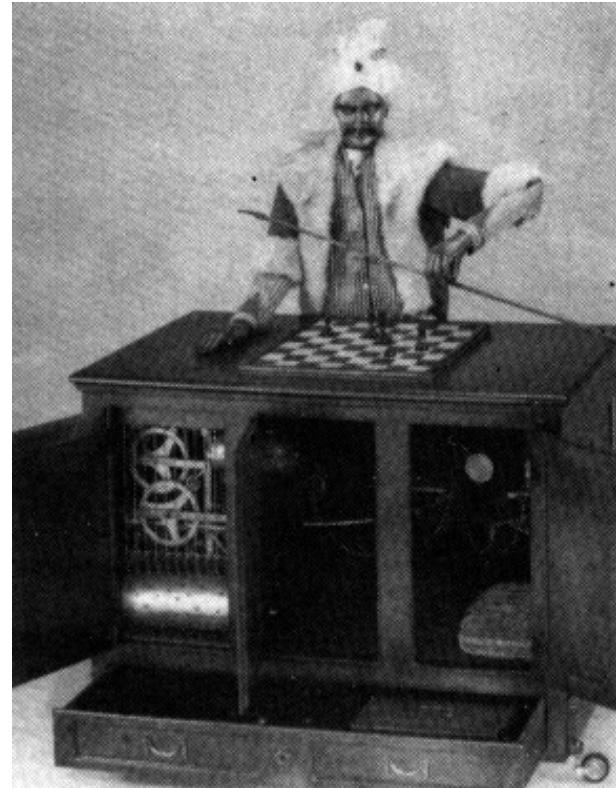
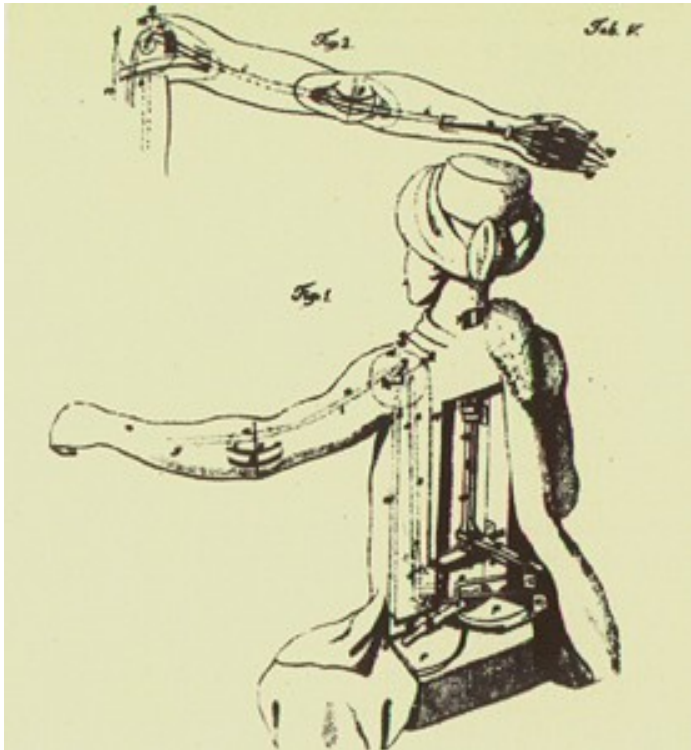
- Zbudował automat szachowy. Ruchy wykonywał ukryty szachista.
- Arcydzieło sztuki



Mephisto – Turek szachista von Kempelena



Mephisto (2)



Abraham Stern (1749–1842)

- Jako pierwszy
skonstruował
kalkulator
wyciągający
pierwiastki

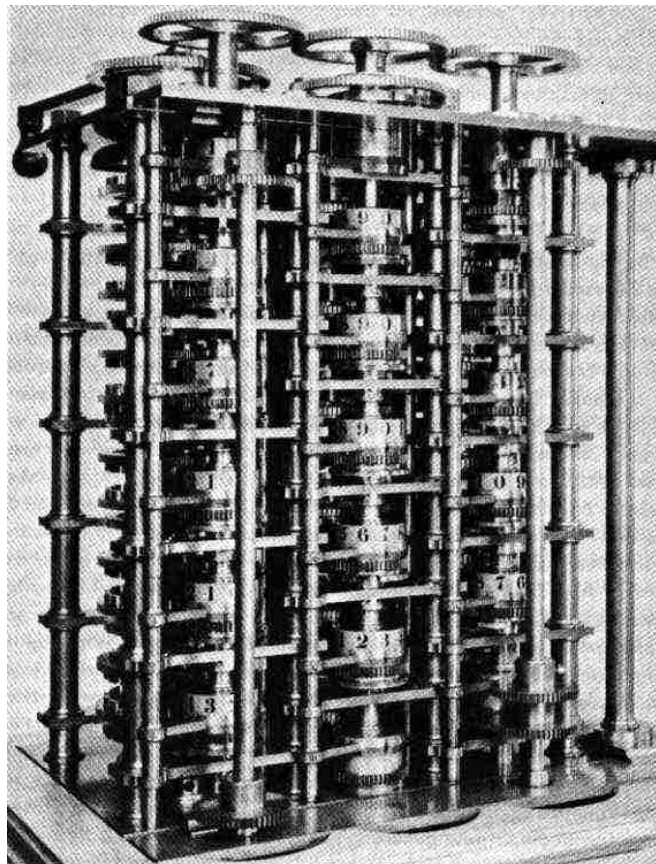


Charles Babbage (1791-1871)

- Twórca pierwszej maszyny liczącej
- Maszyna różnicowa
- Maszyna analityczna



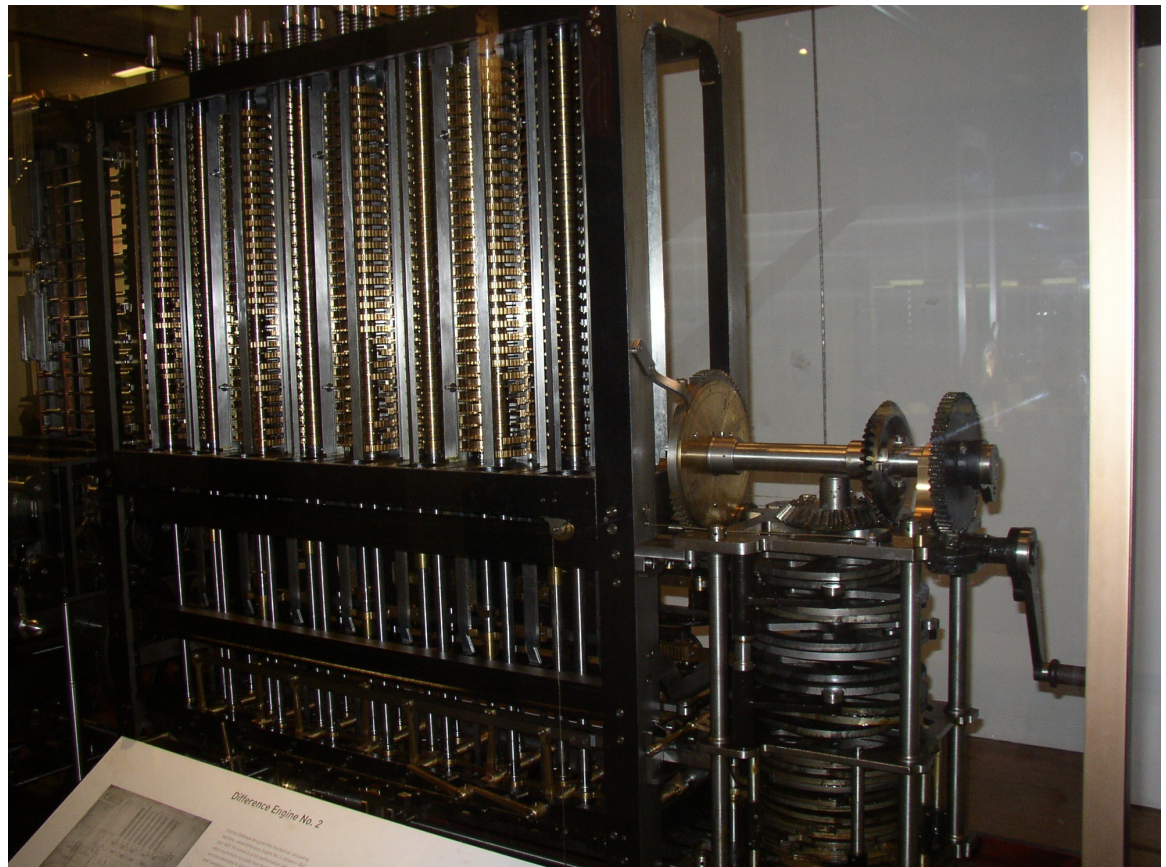
Maszyna różnicowa



Maszyna różnicowa - wersja Babbage'a



Maszyna różnicowa nr 2 - rekonstrukcja z 1990 r.



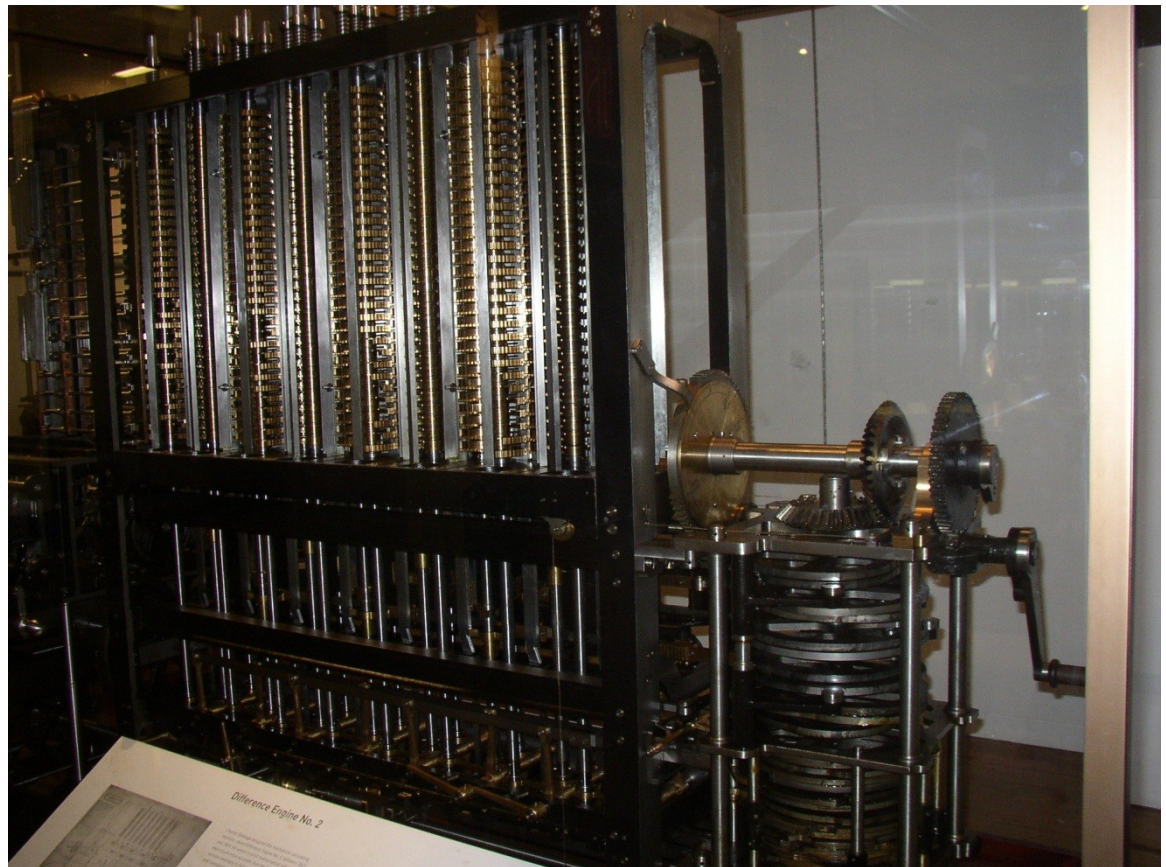
Maszyna różnicowa - wersja Babbage'a

Rekonstrukcja XX-wieczna.

Maszyna służyła do
automatycznego obliczania
wartości wielomianów
metodą różnic skończonych



Maszyna różnicowa nr 2 - rekonstrukcja z 1990 r.



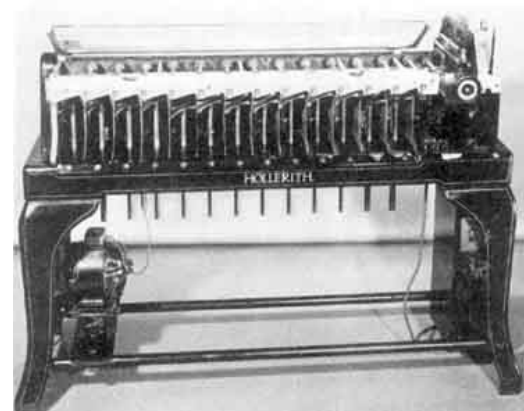
Ada Lovelace (1815-1852)



- Pierwsza programistka
- Opublikowała program obliczający liczby Bernoulliego na (wirtualnej) maszynie analitycznej

Herman Hollerith 1860-1929

- Karty perforowane w przetwarzaniu danych
- Założyciel IBM



Alan Turing (1912-1954) i początki informatyki teoretycznej

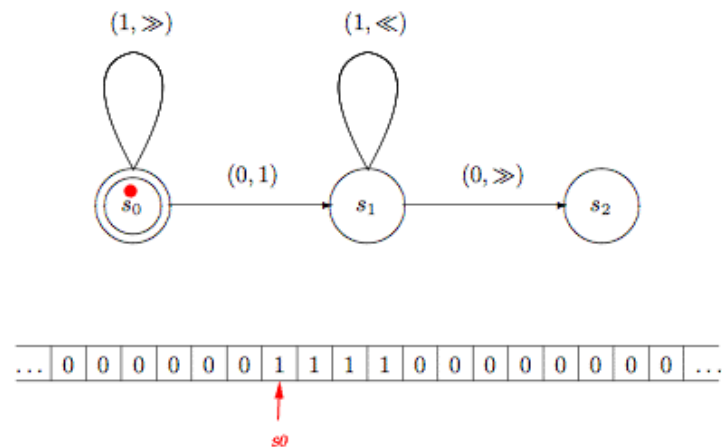
Maszyna Turinga

Składa się z nieskończonej taśmy, głowicy i programu.

Program określa, co w każdym stanie maszyna powinna zrobić w zależności od tego, co jest pod głowicą:

- zapis bitu
- przesunięcie głowicy
- przejście do innego stanu
- zatrzymanie się

Przykładowy program na maszynę Turinga. Dopisanie jedynki.



Etykiety na strzałkach określają do którego stanu przechodzimy i jaką akcję wykonuje głowica. Program z diagramu dopisuje 1 na końcu sekwencji jedynek.

Maszyna Turinga

- Turing wynalazł model niezwykle prosty, a jednocześnie równoważny obliczeniowo współczesnym komputerom: wszystko, co dzisiejsze komputery mogą obliczyć, można zaprogramować na maszynie Turinga.

Czy wszystko da się obliczyć?

- NIE! Okazuje się, że istnieją **problemy nierozstrzygalne**, czyli takie, dla których nie istnieje program je rozwiązujący.
- Przykłady:
 - problem stopu
 - problem odpowiedniości Posta

Problem stopu

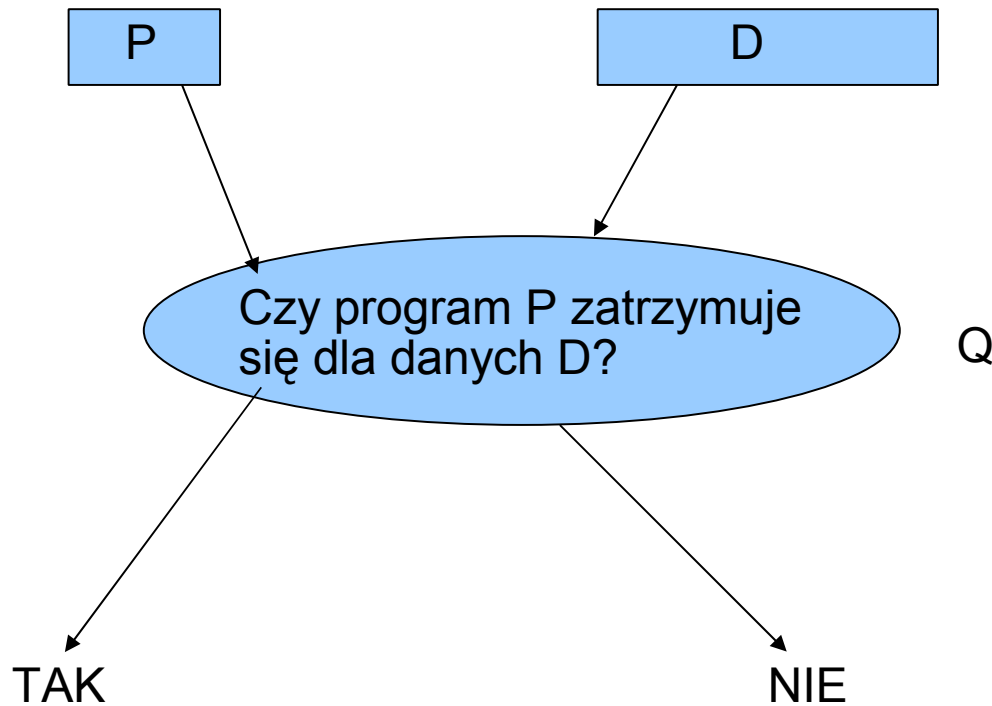
- Czy dla danego programu i dla konkretnych danych na taśmie maszyna Turinga dojdzie do stanu końcowego i się zatrzyma?
- nierozstrzygalność tego problemu oznacza, że nie istnieje program komputerowy, który dla każdego programu P i dla każdych danych D potrafiłby stwierdzić, czy program P zatrzyma się dla danych D , czy się zapętli.

Problem stopu

- Problem stopu: napisz program Q , który dla każdego programu P i danych D rozstrzygnie, czy program P zatrzyma się dla danych D .
- Chodzi o uniwersalną metodę rozstrzygania czy istnieje niebezpieczeństwo zapętlenia się programu.

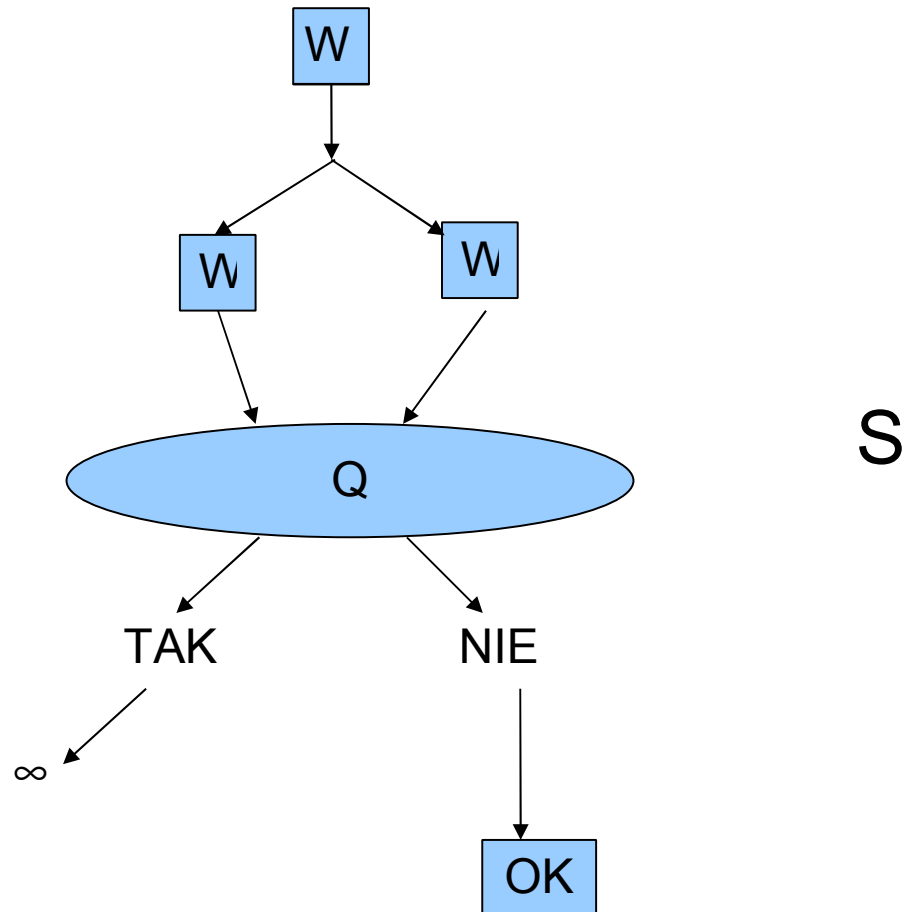
Dowód Turinga

- Chodzi o taki program Q:



Założmy, że taki program Q
istnieje

- Konstruujemy teraz drugi program S:

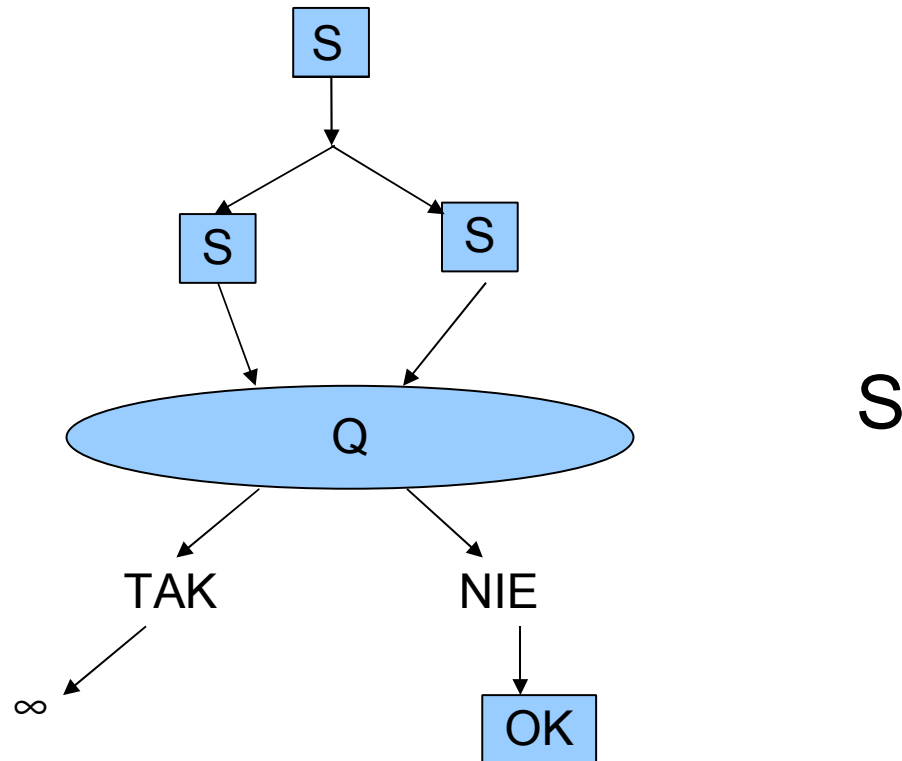


Jak działa program S?

- Program S ma tylko jedną daną W (jakiś program)
- Zaczyna od tego, że robi kopię tej danej
- Następnie uruchamia program Q dla programu W i kodu programu W potraktowanego jako dana.
- Jeśli program W zatrzyma się dla danej W, to program S się zapętla. Jeśli natomiast program W nie zatrzyma się dla danej W, to program S kończy działanie.

Jak działa program S dla danej S?

- Czy program S się zatrzyma dla danej S?



Sprzeczność!

- Okazuje się, że nie sposób odpowiedzieć na pytanie o zatrzymanie się programu S dla danej S .
 - jeśli S się zatrzymuje dla danej S , to Q da odpowiedź TAK i zgodnie z konstrukcją S , program S się zapętli
 - jeśli S się nie zatrzymuje dla danej S , to program Q da odpowiedź NIE i program S się zatrzyma.

Problem odpowiedniości Posta

- Przykład:
 - $x_1=abb$ $y_1=a$
 - $x_2=b$ $y_2=abb$
 - $x_3=a$ $y_3=bb$
- Czy istnieje taki ciąg indeksów $i_1, i_2, \dots, i_n$, że $x_{i_1} \dots x_{i_n} = y_{i_1} \dots y_{i_n}$?
- Problem odpowiedniości Posta jest w ogólnym przypadku nierozstrzygalny! Choć dla niektórych przypadków (np. dla powyższego) można podać odpowiedź, nie ma jednak ogólnego algorytmu, który dla dowolnych danych $x_1, \dots, x_n$ i $y_1, \dots, y_n$ stwierdziłby, czy można wyrównać odpowiednie słowa x-owe i y-kowe za pomocą tego samego ciągu indeksów.

Ciekawy problem nierozstrzygalny

- David Harel w swojej książce "Algorytmika, rzecz o istocie informatyki" podaje przykład ciekawego problemu bazującego na "małpiej" układance.

Małpia układanka

- Mamy skończoną liczbę kwadratowych tabliczek z połówkami małpek. Każda małpka ma górną bądź dolną połowę i kolor.
- Chodzi o to, żeby ułożyć z n^2 zadanych kwadratów kwadrat $n \times n$ tak, żeby wszystkie połówki małpek pasowały do siebie. .
- Na kolejnym slajdzie jest przykładowa instancja tego problemu dla $n=3$ i małpka = piesek.



Matpia układanka 3x3

- To ułożenie z poprzedniego slajdu jest złe - wszystkie pieski powinny mieć połówki tego samego koloru i różnych typów (tylna, przednia), a na zdjęciu nie mają.
- Ciekawe uogólnienie tego problemu jest dla nieskończonej płaszczyzny.

Nieskończona matpia układanka

- Nieskończona wersja tej łamigłówki ma nieskończoną liczbę kopii każdego ze skończonej liczby kartoników. Pytania:
 - Czy z zadanego zestawu kartoników da się ułożyć płaszczyznę?
 - Czy z zadanego zestawu kartoników da się ułożyć półpłaszczyznę?

Nieskończona matpia układanka

- Odpowiedź na to pytanie może zaskakiwać:
 - Problem dla płaszczyzny jest dość prosty
 - Problem dla półpłaszczyzny jest nierozstrzygalny!

Pierwsze prawdziwe komputery

Twórcy pierwszych komputerów

Konrad Zuse

John Atanasoff

Howard Aiken

John Mauchly &
J. Presper Eckert

Konrad Zuse

Niemcy, 1939



“Byłem zbyt leniwy, aby trudzić się obliczeniami, więc wymyśliłem komputer.”

Komputer Zusego działał za pomocą elektrycznych przełączników, zastąpionych później lampami.

Z1 - replika z Deutsches Museum (Monachium)



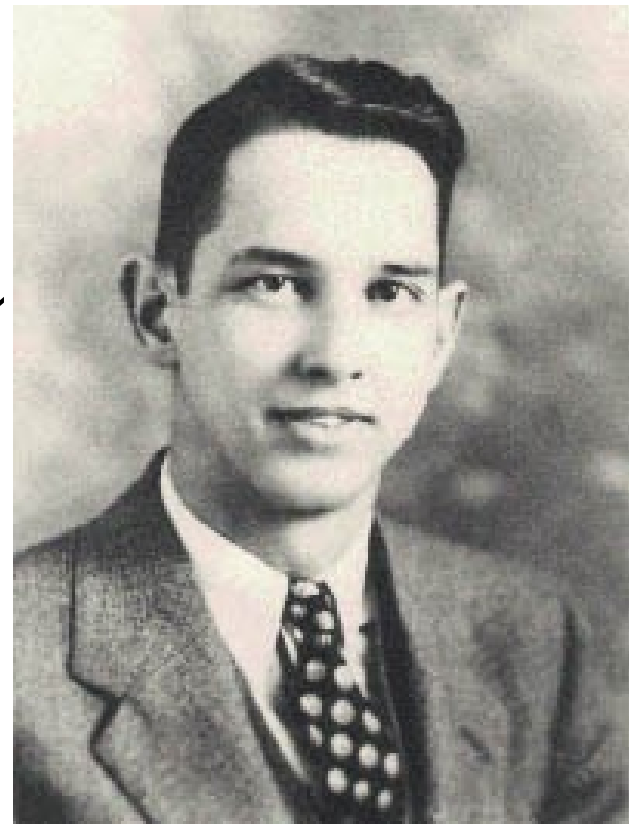
John Atanasoff

USA, 1939

Atanasoff-Berry Computer
(ABC)

Komputer ABC był już w
technologii lampowej i działał
w arytmetyce binarnej

Nigdy nie został skończony.



Howard Aiken

USA, 1944

Mark 1 był największym komputerem który kiedykolwiek został zbudowany!

Działał w technologii przełącznikowej i przyjmował instrukcje wprowadzane za pomocą taśmy perforowanej.



John Mauchly oraz Presper Eckert

USA, 1945



Mauchly oraz Eckert
zbudowali ENIACa
(Electronic Numerical
Integrator and Computer).

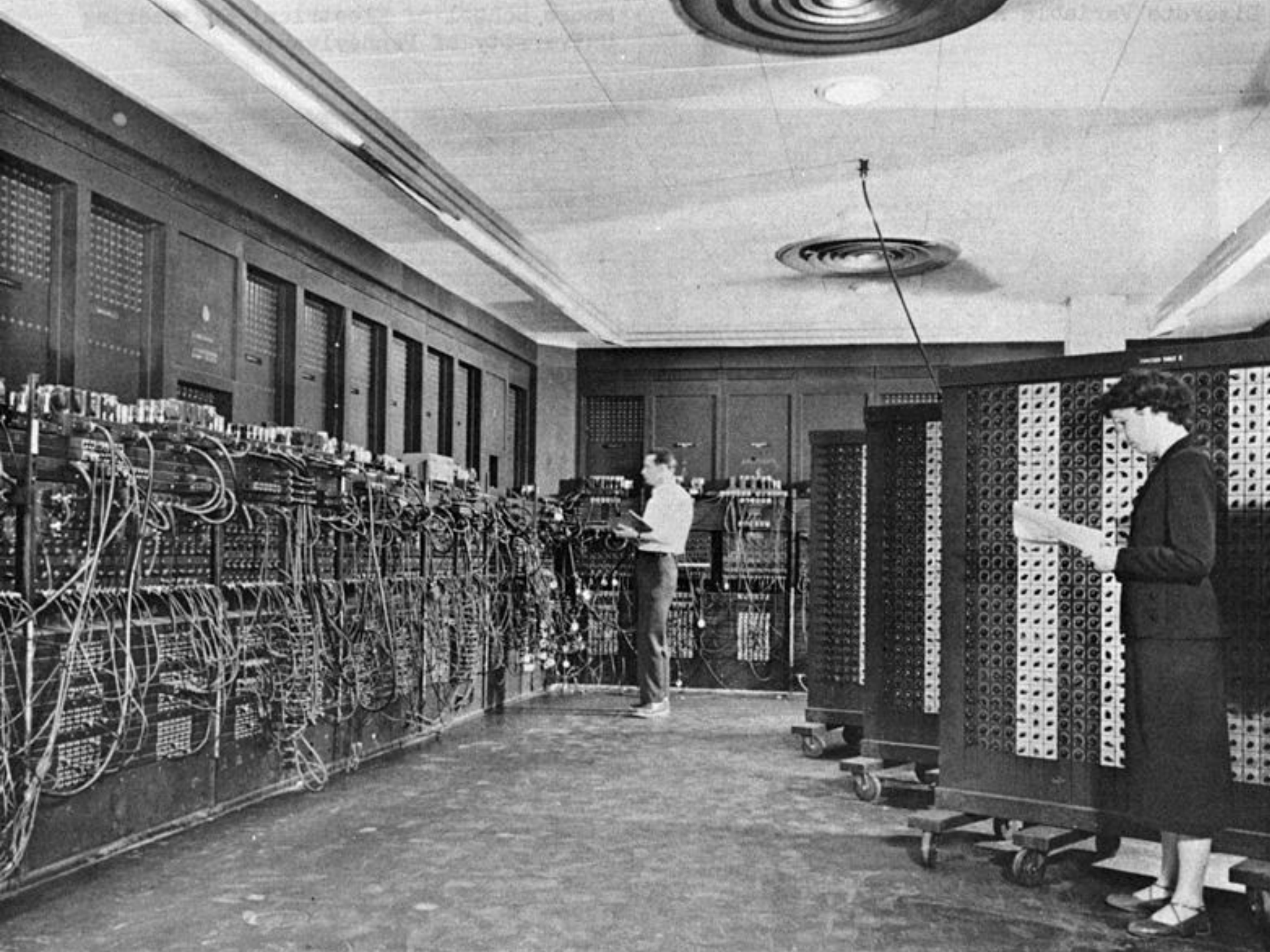
ENIAC był zbudowany w
technologii lampowej i
programowany za pomocą
zestawiania obwodów.

John von Neumann (1903-57)

John von Neumann – jeden z najgenialniejszych matematyków wszystkich czasów.

Wspólnie z Mauchlym i Eckertem zaproponowali architekturę, w której pamięć danych i rozkazów jest współdzielona, a procesor pracuje w tzw. cyklu von Neumanna wykonując kolejno odczyt instrukcji, jej interpretację, wykonanie i aktualizację licznika rozkazów.







Assembly

- Programowanie w języku maszynowym było horrendalnie trudne. Assembler wprowadził
 - Symboliczne adresy
 - Symboliczne zmienne
 - Makra

John Backus (1924-2007)

- FORTRAN (1955) był pierwszym językiem programowania z prawdziwego zdarzenia
- Miał zmienne, symboliczne adresy, łatwe zapisywanie wyrażeń, instrukcje warunkowe, pętle, procedury, ciekawe zarządzanie pamięcią



Algol 60

- W końcu lat 50-tych międzynarodowy zespół zaprojektował język nowej generacji, który stał się protoplastą całej serii języków takich jak Euler, Pascal, C, Simula.
- ALGOL 60 posiadał składnię i semantykę opisaną za pomocą gramatyk Chomsky'ego

ALGOL 60

- Algol 60 było językiem strukturalnym. Zawierał:
 - Instrukcję przypisania
 - Instrukcję warunkową
 - Instrukcję pętli
 - Mechanizm blokowania
 - Mechanizm procedur (rekursja!)

Niklaus Wirth - PASCAL



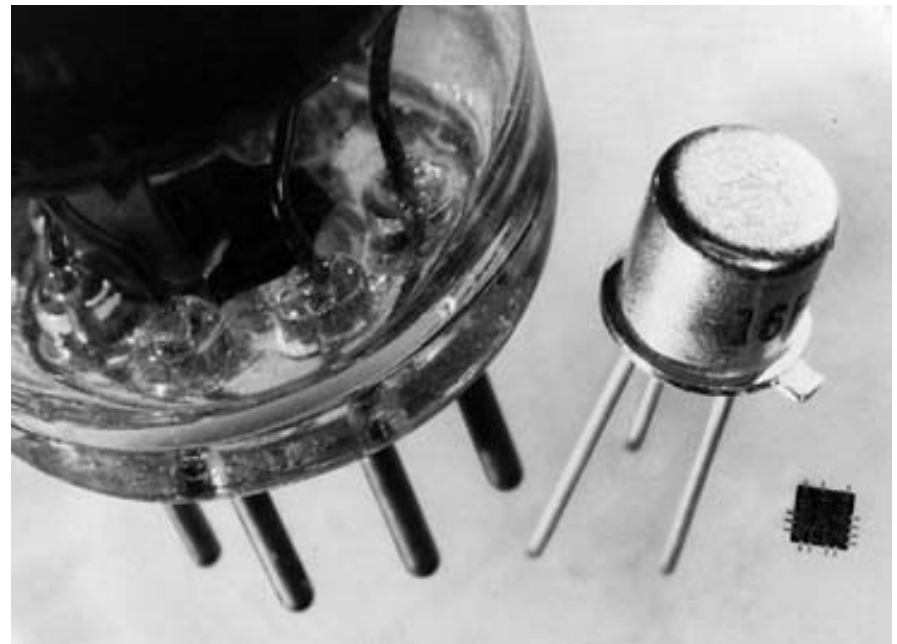
- N. Wirth - twórca PASCALA (1970)
- Język przeznaczony do dydaktyki
- Kompilator napisany w PASCALU tłumaczył programy na P-code, czyli assembler wirtualnej maszyny łatwej do zaimplementowania na każdym komputerze.

PASCAL

- Specyficzne cechy wprowadzone w Pascalu:
 - Możliwość definiowania własnych typów
 - Zmienne wskaźnikowe
 - Rekordy
 - Instrukcja wyboru (case)
 - Parametry wołane przez wartość i przez zmienną

Rozwój i przyspieszenie

- Pierwsza generacja
 - Lampy próżniowe
- Druga generacja
 - Tranzystory
- Trzecia generacja
 - Układy scalone
- Czwarta generacja
 - Mikroprocessory



Pierwsza generacja komputerów

- Lata 30-te i 40-te
- Lampy próżniowe w roli przełączników
- Duże komputery
- Niezwykle powolne, jak na dzisiejsze standardy
- Podatne na błędy
- ABC, Mark I, ENIAC, UNIVAC i in

Druga generacja

- Lata 50-te do połowy 60-tych
- Tranzystory w roli przełączników
- Znacznie mniejsze niż lampowe
- Mniej więcej tysiąckrotnie szybsze
- Tańsze i pewniejsze

Trzecia generacja

- Późne lata 60-te
- Krzemowe czipy w roli przełączników
- Znaczne obniżenie kosztu i rozmiarów
- Istotny wzrost szybkości i wydajności

Czwarta generacja

- Lata 70-te do dziś
- Zestawy przełączników zastąpione jednym mikroprocesorem
- Cena tak spadła, że stały się dostępne powszechnie

