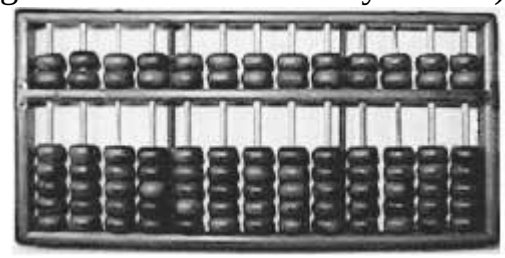
Jak to wcześniej bywało, czyli wyjątki z historii maszyn algorytmicznych

Algorytmika jest ściśle związana z rewolucją techniczną dokonującą się na przestrzeni zaledwie ostatnich dwustu lat. Owszem, jeśli zechcemy traktować informatykę i algorytmikę jako pewną całość wywodzącą się naturalnie z systemów obliczeniowych, to warto wspomnieć o osiągnięciach ludów sumeryjskich, wynalazców tabliczek obliczeniowych, własnego kalendarza i sześćdziesiętnego systemu pomiarowego (24-godzinna doba to ich wynalazek). Znani są też Chińczycy, wynalazcy Abakusa, czyli najsłynniejszego liczydła w historii ludzkości, choć mało kto ma świadomość, że praktycznie każdy w miarę cywilizowany lud dopracowywał się jakiegoś systemu wspomagającego obliczenia i trudno tu przyznawać komuś palmę pierwszeństwa. Ponadto, licytując się tego typu faktami, łatwo cofniemy się do okresu datowanego na 2 - 3 tysiące lat p.n.e., tylko czy to ma obecnie wartość inną niż ciekawostki?



Aby nie zamieniać tej prezentacji w podręcznik historii, pominę rozważania na temat maszyny do dodawania Blaise'a Pascala (ok. roku 1645) lub jej niemieckiego odpowiednika skonstruowanego nieco później przez Gottfrieda Wilhelma Leibniza. Popatrzmy jedynie na kilka charakterystycznych wydarzeń związanych z wynalazkami, które nie tylko ułatwiały obliczanie, ale i pozwalały na elementarne programowanie, czyli coś, co już jednoznacznie kojarzy się z komputerami i algorytmami.

—1801 —

Francuz Joseph Marie Jacquard wynalazł krosno tkackie, w którym wzorzec tkaniny był „programowany" na swego rodzaju kartach perforowanych. Proces tkania był kontrolowany przez algorytm zakodowany w postaci sekwencji otworów wybitych w karcie. Sam pomysł stanowił wynik wielu lat pracy Jacquarda i mógł ujrzeć światło dzienne dzięki przypadkowi, jakim było uczestnictwo w konkursie państwowym, na którym przedstawił maszynę do robienia sieci rybackich.

Koncepcja Jacquarda[1] zainteresowała francuskiego matematyka, Lazare Carnota, który ściągnął go do Paryża w celu kontynuowania badań i pomógł w uzyskaniu stypendium rządowego. Pierwsze prace omal nie doprowadziły do śmierci wynalazcy, rozwścieczeni tkacze niemal utopili go w Rodanie, przeczuwając, że jego maszyna stanowi zagrożenie dla ich zatrudnienia (a dokładniej dla zatrudnienia ich dzieci, które do tej pory służyły za pomocników podnoszących nitki, aby umożliwić utkanie lub nie odpowiedniego wzoru przez przesuwającą się poprzecznie cewkę z nitką — wynalazek Jacquarda eliminował pięć stanowisk pracy przy jednym krośnie!). Pomysł Jacquarda był dopasowany do ówczesnych możliwości technicznych, ale warto zauważyć, że karta perforowana z zakodowaną logiką dwustanową (dziurka lub jej brak oznaczał dla maszyny tkackiej podjęcie lub nie odpowiedniej akcji natury mechanicznej) jest prekursorem współczesnych pamięci, w których za pomocą liczb binarnych koduje się odpowiednie akcje algorytmu!

— 1833 —

Anglik Charles Babbage częściowo zbudował maszynę do wyliczania niektórych formuł matematycznych. W czasach, w których żył Babbage, nastąpiła eksplozja zastosowań matematyki (astronomia, nawigacja), a nie istniały metody wspomagające obliczenia w sposób automatyczny. Babbage był autorem koncepcji tzw. maszyny analitycznej, zbliżonej do swego poprzedniego dzielą ale wyposażonej w możliwość przeprogramowywania, jak w przypadku maszyny Jacquarda.

[1] Na początku lat 1800-nych Francuz zajmujący się konstruowaniem krosien dla przemysłu tkackiego i noszący nazwisko Joseph-Marie Jacquard wynalazł metodę automatycznego sterowania układem wątku i osnowy nici na krosnach jedwabnych, która polegała na zapisie wzorów dziurek na zestawie specjalnych kart z wybitymi otworami..

W pewnym uproszczeniu maszyna ta miała składać się z magazynu (odpowiednika pamięci realizowanego jako kolumny kół, później zastąpionego bębnem), młyna (jednostki obliczeniowej wykonującej operacje za pomocą obrotów kół i przekładni) i mechanizmu sterującego wykorzystującego karty perforowane (Jacquard!). Czyż nie przypomina to schematu komputera?

Opisy maszyny Babbage'a były na tyle dokładne, że matematyczka Ada Lovelace, córka lorda Byrona, opracowała pierwsze teoretyczne „programy" dla tej nieistniejącej jeszcze maszyny, stając się pierwszą uznaną... programistką w historii informatyki[2]!

Wymagania natury mechanicznej, jakie stawiała ta maszyna, pozwoliły na skonstruowanie jej pierwszego prototypu dopiero w dwadzieścia lat od narodzin samej idei, a sama maszyna powstała dopiero w roku... 1992, oczywiście bardziej jako ciekawostka niż potrzeba naukowa.

— 1890 —

W zasadzie pierwsze publiczne i na dużą skalę użycie maszyny bazującej na kartach perforowanych. Chodzi o maszynę do opracowywania danych statystycznych, dzieło Amerykanina Hermana Holleritha, użyte podczas spisu ludności. Na marginesie warto dodać, że przedsiębiorstwo Holleritha przekształciło się w 1911 roku w International Business Machines Corp., bardziej znane jako IBM, do dziś czołowego producenta komputerów.

— lata trzydzieste XX wieku —

Rozwój badań nad teorią algorytmów (plejada znanych matematyków: Turing, Gödel, Markow). Z tego okresu wywodzi się słynne zagadnienie postawione przez pruskiego[3] matematyka Dawida Hilberta, który w 1928 roku na międzynarodowym kongresie matematyków publicznie postawił pytanie, czy

[2] Od jej imienia pochodzi nazwa języka programowania ADA.

[3] Urodzony w Königsbergu, obecnie zwanym Kaliningradem.

istnieje metoda pozwalająca rozstrzygnąć prawdziwość dowolnego twierdzenia matematycznego, w wyniku li tylko mechanicznych operacji na symbolach. Studentom informatyki bliskie będzie pojęcie tzw. maszyny Turinga, abstrakcyjnej maszyny obliczeniowej złożonej z głowicy czytająco-piszącej oraz nieskończonej taśmy zawierającej symbole (np. liczby lub operatory działań). Ten abstrakcyjny model matematyczny stworzył podwaliny pod współczesne komputery. Uważam, że powinieneś zapamiętać tylko, że to, co określa się nieco myląco terminem maszyna, jest wyłącznie modelem schematu działania wg zadanego algorytmu.

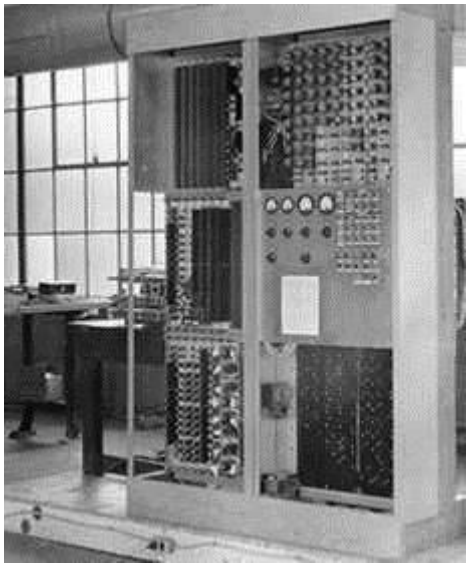
W latach trzydziestych XX wieku możemy też zaobserwować jedno z pierwszych niechlubnych zastosowań komputerów: do niemieckiej III Rzeszy (a później także do państw okupowanych przez hitlerowców) trafiają tysiące maszyn sortujących IBM. Wspomagając spis ludności, pozwoliły utworzyć rejestry osób przeznczonych do likwidacji (np. upośledzonych czy pochodzenia żydowskiego)[4].

— lata czterdzieste XX wieku —

Budowa pierwszych komputerów ogólnego przeznaczenia (głównie dla potrzeb obliczeniowych wynikłych w tym wojennym okresie, czyli badań nad łamaniem kodów, początku „kariery" bomby atomowej).

Pierwszym urządzeniem, które można określić jako „komputer", był automatyczny kalkulator MARK 1, skonstruowany w 1944 roku (jeszcze na przekaźnikach, czyli jako urządzenie elektromechaniczne). Jego twórcą był Amerykanin Howard Aiken z uniwersytetu Harvarda. Aiken bazował na idei Babbage'a, która musiała czekać 100 lat na swoją praktyczną realizację! Dwa lata później powstaje pierwszy elektroniczny komputer ENIAC[5] (jego wynalazcy to John Presper Eckert i John William Mauchly z uniwersytetu Pensylwania), który miał oryginalnie wspomagać obliczenia balistyczne.

[4] Źródło: Edwin Black, IBM i holocaust. Strategiczny sojusz hitlerowskich Niemiec z amerykańską korporacją.
[5] Ang. Electronic Numerical Interpreter And Calculator.



Powszechnie jednak za pierwszy komputer w pełnym tego słowa znaczeniu uważa się EDVAC[6] zbudowany na uniwersytecie w Princeton. Jego wyjątkowość polegała na umieszczeniu programu wykonywanego przez komputer całkowicie w jego pamięci, podobnie jak i pamięci do przechowywania wyników obliczeń. Autorem tej przełomowej idei był matematyk Johannes von Neumann (Amerykanin węgierskiego pochodzenia)[7].

— okres powojenny —

Prace nad komputerami prowadzone są w wielu krajach równolegle. W grę zaczyna wchodzić wkroczenie na nowo powstały obiecujący rynek komputerów (gdyż kończy się era budowania unikalnych uniwersyteckich prototypów). Na rynku pojawiają się kalkulatory IBM 604 i BULL GAMMA 3, a następnie duże komputery naukowe, np. UNIVAC 1 i IBM 650. Zaczynającej się zarysowywać dominacji niektórych producentów usiłują przeciwdziałać badania prowadzone w wielu krajach (mniej lub bardziej systematycznie i z różnorakim poparciem polityków), ale... to już jest temat na osobną książkę!

—1969 —

Rok 1969 można uznać za kamień milowy w rozwoju systemów operacyjnych i języków programowania. Ken Thompson i Dennis Ritchie opracowują koncepcję systemu UNIX, prekursora nowoczesnych systemów operacyjnych (to wówczas zrodziła się idea systemu plikowego i szereg innych koncepcji obecnych do dziś na Twoim pececie lub maku). Pierwsza dojrzała wersja systemu UNIX powstała w latach 1969 - 1973 i jego końcowe wydanie zostało napisane z użyciem języka C, który stał się w tym czasie głównym językiem programowania i narzędziem do... tworzenia innych języków programowania (gwoli ścisłości, kompilatorów, a nie samych języków). Język C jest także prekursorem używanego w tej książce C++. Język C pozwolił wreszcie wyzwolić się z programowania sprzętowego, można było zacząć pisać programy w języku wysokiego poziomu, bez wiązania

[6] Ang. Electronic Discrete Variable Automatic Computer.

[7] Koncepcja komputera została opracowana w 1946 roku, jednak jego pierwsza realizacja praktyczna powstała dopiero w roku 1956.

się z konkretnym sprzętem (taka zależność funkcjonuje dla języków typu asembler, gdzie musisz znać instrukcje procesora i szczegóły modelu pamięci).

— teraz —

Burzliwy rozwój elektroniki powoduje masową do dziś trwającą komputeryzację wszelkich dziedzin życia. Komputery stają się czymś powszechnym i niezbędnym, wykonując takie różnorodne zadania, jakie tylko nakaże im ludzka wyobraźnia. Rozpowszechnienie komputerów spowodowało duży nacisk na jakość ich działania: programiści w firmach komputerowych systematycznie używają technik wspomagających tworzenie kodu dobrej jakości. Oto te techniki.

♦ Systemy kontroli wersji, pozwalające na zarządzanie wersjami plików źródłowych, takie jak śledzenie zmian, identyfikacja autorów poprawek, odtwarzanie wersji poprzednich, budowanie tzw. wydań, czyli integralnych, dających się instalować wersji nowych systemów.

♦ Automatyczne testowanie, przydające się zwłaszcza podczas tworzenia dużych systemów komputerowych, tworzonych przez wielu różnych programistów.

♦ Ciągła integracja (ang. continuous integration) polegająca na kontrolowanym, stałym włączaniu zmian w kodzie do wersji końcowej systemu, co ułatwia wykrywanie błędów jeszcze przed fazą oficjalnych testów systemowych.

♦ Stosowanie tzw. wzorców projektowych, czyli szablonów ułatwiających implementację często pojawiających się problemów projektowych.

♦ Architektury warstwowe (np. rozdzielenie danych od ich prezentacji oraz od logik biznesowych, które nimi „manipulują").

♦ Podejście komponentowe i stosowanie tzw. interfejsów, czyli ukrywanie realizacji algorytmów od ich używania.
Jak to się niedawno odbyło, czyli o tym, kto „wymyślił" metodologię programowania

Zamieszczone w poprzednim paragrafie kalendarium zostało doprowadzone do momentu, w którym programiści zaczęli mieć do dyspozycji komputery z prawdziwego zdarzenia. Olbrzymi nacisk, jaki był kładziony na rozwój sprzętu, w istocie doprowadził do znakomitych rezultatów — efekt jest widoczny dzisiaj w praktycznie każdym biurze i w coraz większej liczbie domów prywatnych.

W latach sześćdziesiątych XX wieku zaczęto konstruować pierwsze naprawdę duże systemy informatyczne — w sensie ilości kodu, głównie asemblerowego, wyprodukowanego na poczet danej aplikacji. Ponieważ jednak programowanie było ciągle traktowane jako działalność polegająca głównie na intuicji i wyczuciu, zdarzały się całkiem poważne wpadki w konstrukcji oprogramowania: albo szybko tworzone były systemy o małej wiarygodności, albo też nakład pieniędzy włożonych w rozwój produktu znacznie przewyższał szacowane wydatki i stawiał pod znakiem zapytania sens podjętego przedsięwzięcia. Brakowało zarówno metod, jak i narzędzi umożliwiających sprawdzanie poprawności programowania. Powszechną metodą programowania było testowanie programu aż do momentu jego całkowitego „odpluskwienia”[8]. Warto zwrócić jeszcze uwagę, że oba wspomniane czynniki, czyli wiarygodność systemów i poziom nakładów, są niezmiernie ważne w praktyce; informatyczny system bankowy musi albo działać stuprocentowo dobrze, albo nie powinien być w ogóle oddany do użytku! Z drugiej strony poziom nakładów przeznaczonych na rozwój oprogramowania nie powinien odbić się niekorzystnie na kondycji finansowej przedsiębiorstwa.

W pewnym momencie sytuacja stała się tak krytyczna, że zaczęto wręcz mówić o kryzysie w rozwoju oprogramowania! W 1968 roku została nawet zwołana konferencja NATO (Garmisch, Niemcy) poświęcona przedyskutowaniu zaistniałej sytuacji. Rok później w ramach IFIP (ang. International Federation for Information Processing) została utworzona specjalna grupa robocza pracująca nad tzw. metodologią programowania.

Z historycznego punktu widzenia dyskusja na temat udowadniania poprawności algorytmów zaczęła się jednak od artykułu Johna McCarthy’ego „A basis for a mathematical theory of computation”, gdzie padło zdanie: „zamiast sprawdzania programów komputerowych metodą prób i błędów aż do momentu ich całkowitego odpluskwienia powinniśmy udowadniać, że posiadają one pożądane własności”. Nazwiska ludzi, którzy zajmowali się teoretycznymi pracami na temat metodologii programowania, nie znikły z horyzontu; są to Dijkstra, Hoare, Floyd, Wirth itd.

Warto również zauważyć, że algorytmika nie jest to nauka, która powstała samorodnie. O ile obecnie należy ją odróżniać jako odrębną gałąź wiedzy, to nie sposób nie docenić wielowiekowej pracy matematyków, którzy dostarczyli

[8] Żargonowe określenie procesu usuwania błędów z programu.

algorytmice zarówno narzędzi opisu zagadnień, jak i wielu użytecznych teoretycznych rezultatów.