

## Informatyka,

dyscyplina nauk. zajmująca się przetwarzaniem informacji z użyciem komputerów; także dziedzina działalności gospodarczej związana z wykorzystaniem komputerów, zwłaszcza z wytwarzaniem oprogramowania i świadczeniem usług (przemysł informatyczny).

Podstawę dla informatyki stanowią różne teorie matematyki, przede wszystkim teoria informacji, określająca możliwe sposoby zapisu informacji w postaci cyfrowej, w szczególności w postaci ciągów bitów, z których każdy może przyjmować wartości 0 lub 1 (teoretycznie jest możliwe budowanie komputerów realizujących logikę wielowartościową); podstawę do opisu architektury, czyli logicznej struktury komputerów stanowi algebra Boole'a, określająca możliwe operacje na pojedynczych bitach oraz na ich ciągach. W zakres informatyki nie wchodzi wiedza dotycząca samej budowy komputerów — w sferze elektroniki leżą problemy związane ze sposobem reprezentowania bitów w komputerze oraz techniczne rozwiązania układów wykonujących poszczególne operacje.

**Informatyka jako nauka** obejmuje teoretyczne podstawy informatyki (w tym teorię obliczeń i analizę algorytmów), ściśle związane z matematyką, ze szczególnym uwzględnieniem lingwistyki matematycznej (→ język formalny) i teorii automatów (→ automat abstrakcyjny), a także m.in. teorii → grafów, oraz naukę o efektywnym pisaniu programów, związaną z konkretnymi ich klasami, np. → kompilatorami, → systemami operacyjnymi, → bazami danych, systemami → doradczymi.

Teoria obliczeń obejmuje badania związane z definiowaniem i analizowaniem problemów, które mogą być rozwiązywane w sposób algorytmiczny, oraz badaniem czasu potrzebnego do ich rozwiązania (→ złożoność obliczeniowa); tematyka ta jest ściśle powiązana z logiką matematyczną, a także teorią funkcji rekurencyjnych. Istnieje wiele równoważnych modeli, pozwalających opisać obliczenia w sposób formalny (np. → Turinga maszyna, → rachunek  $\lambda$  Churcha, semi-systemy Thuego). Do opisu zachowania się programów zapisanych w językach programowania powstały specjalne systemy logiki, np. logika dynamiczna, logika temporalna, logika Hoare'a, logika wielowartościowa i logika algorytmiczna. Istotne wyniki dotyczą niemożliwości rozwiązania w sposób algorytmiczny pewnych problemów, np. zbudowania algorytmu określającego, czy dowolny algorytm zakończy pracę w skończonym czasie dla dowolnych danych (nierozstrzygalność, częściowa rozstrzygalność, problem stopu). Analiza algorytmów obejmuje badania własności algorytmów przetwarzania danych, w szczególności ich poprawności, złożoności i efektywności dla różnych danych. Dla komputerów opracowano wiele algorytmów znacznie szybszych niż wynikające bezpośrednio z definicji matematycznej, np. szybki algorytm transformacji Fouriera w  $n$  punktach (FFT, → Fouriera szybka transformata) o koszcie rzędu  $n \cdot \log n$  mnożeń (tradycyjny algorytm wymaga  $n \cdot n$  mnożeń); użycie FFT znacznie zwiększyło możliwości zastosowania komputerów do analizy sygnałów (np. dla  $n = 10^6$  pozwala skrócić czas obliczeń ok. 50 tys. razy); także algorytm (Strassena) mnożenia macierzy o wymiarze  $n \times n$ , wymagający ok.  $n^{\log 7}$  mnożeń (ok.  $n^{2,81}$ , tradycyjny algorytm wymaga  $n^3$  mnożeń); jego użycie pozwala zmniejszyć koszt rozwiązywania równań liniowych metodą wzorowaną na eliminacji Gaussa. Dla praktycznych zastosowań w zakresie algebry liniowej sformułowano algorytmy iteracyjne, o jeszcze mniejszej złożoności. Opisano algorytmy sortowania  $n$  liczb o złożoności rzędu  $n \cdot \log n$  porównań (złożoność tradycyjnych algorytmów wynosi  $n^2$  porównań). Problemy, dla których istnieją algorytmy o złożoności wielomianowej, należą do tzw. klasy P (ang. *polynomial*). Dla niektórych problemów (należących do tzw. klasy NP — ang. *non-deterministic polynomial*) są znane wyłącznie algorytmy bardzo nieefektywne, o złożoności wykładniczej (rzędu  $2^n$ , gdzie  $n$  jest liczbą określającą rozmiar danych); niektóre z tych problemów są między sobą równoważne (m.in. sprawdzenie, czy formuła boolowska rachunku zdań z  $n$  zmiennymi może być prawdziwa, problem cyklu Hamiltona i zadanie

→ komiwożera z teorii grafów); znalezienie algorytmu o złożoności wielomianowej dla jednego z nich umożliwiłoby skonstruowanie algorytmów dla pozostałych (tzw. problemy NP-zupełne). Istnieje hipoteza, że algorytmy o złożoności wielomianowej dla tych problemów nie istnieją (→ hipoteza  $P \neq NP$ ). Przy analizie poprawności algorytmów formułuje się twierdzenia o zachowaniu programu (np. o zależnościach między wartościami zmiennych po wykonaniu pewnej instrukcji programu), a następnie stosuje się pewne techniki stopniowego dowodzenia tych twierdzeń, wzorowane na regułach wnioskowania w logice. Specjalne techniki są stosowane do analizy algorytmów współbieżnych, np. sieci → Petriego.

**Architektura komputerów** obejmuje badania poświęcone logicznej strukturze komputerów, niezależnie od jej fizycznej realizacji; wykorzystuje wyniki analizy algorytmów, np. w zakresie projektowania najefektywniejszych układów wykonywania operacji arytmetycznych (szczególnie mnożenia i dzielenia). Obecnie badania koncentrują się nad analizą systemów równoległych, zbudowanych z tysięcy → procesorów współpracujących przy rozwiązywaniu złożonych problemów.

**Badania** dotyczące zasad konstruowania języków programowania dotyczą sposobu definiowania składni tych języków oraz ich semantyki (precyzyjnej interpretacji poszczególnych wyrażeń i instrukcji); do opisu składni korzysta się zwykle z → gramatyk formalnych, dobranych tak, aby było możliwe ich efektywne rozpoznawanie; w ogólnym przypadku gramatyk bezkontekstowych koszt analizowania programu może rosnąć wykładniczo wraz ze wzrostem długości programu; praktycznie stosuje się pewne podklasy gramatyk bezkontekstowych, dla których koszt analizy rośnie liniowo. Do opisu semantyki języka programowania stosuje się techniki matematyczne, np. semantyka denotacyjna definiuje program jako funkcję powstałą ze złożenia funkcji odpowiadających poszczególnym instrukcjom, a semantyka operacyjna definiuje jednoznacznie sposób tłumaczenia instrukcji języka na ciąg instrukcji pewnego hipotetycznego komputera. Podejmowano próby takiego definiowania języków programowania, żeby ułatwić proces weryfikowania poprawności programów zapisanych w tych językach i zapewnić narzędzia wspomagające programistę przy dokonywaniu takiej analizy (w pełni automatyczne weryfikowanie jest niemożliwe ze względu na nierozstrzygalność problemu).

**Badania systemów operacyjnych** dotyczą sposobu realizowania obsługi użytkownika oraz innych programów, tak aby zapewnić im bezpieczeństwo i wygodę, a równocześnie efektywnie wykorzystywać zasoby całego systemu informatycznego. Badane problemy są związane m.in. ze zjawiskami pojawiającymi się przy → współbieżności i dostępie do dzielonych zasobów, np. → plików lub → drukarek (zagłódzenie, blokada). Badania efektywności dotyczą np. sposobu zapisywania plików na → dyskach, tak aby efektywnie wykorzystywać powierzchnię dysków, a jednocześnie umożliwić szybki dostęp do zapisanej informacji. Obecnie szczególną wagę przykładają się do badań nad systemami operacyjnymi dla komputerów wyposażonych w wiele procesorów oraz dla systemów rozproszonych, składających się z komputerów połączonych → siecią komputerową. Techniki takie, jak zdalne wywoływanie procedur oraz rozproszone systemy plików pozwalają wykorzystać dużą moc obliczeniową takich systemów, bez konieczności modyfikowania używanych programów. Systemy operacyjne dla komputerów połączonych ze sobą w tzw. klastry zapewniają prawidłową pracę programów nawet w przypadku awarii jednego z komputerów klastra.

**Badania baz danych** dotyczą zarówno log. modelu bazy danych, czyli sposobu reprezentowania informacji o log. zależnościach między zapisanymi danymi, jak i sposobu efektywnego zapisywania informacji na fiz. nośnikach danych oraz dostępu do nich. Obecnie przedmiotem badań są bazy zawierające informacje multimedialne (np. baza filmów, zapisanych w postaci cyfrowej, dostępnych przez sieć komputerową) oraz hurtownie danych, zawierające wielkie ilości szczegółowych danych, służących do analiz statystycznych (np. baza wszystkich indywidualnych zakupów w supermarkecie służąca do analizy, jak rozmieszczenie produktów na półkach wpływa na ich sprzedaż); trwają prace nad rozszerzeniami języka → SQL, umożliwiającymi wygodny i efektywny dostęp do takich baz.

**Badania sieci komputerowych** obejmują opracowywanie i analizę protokołów komunikacyjnych, dostosowanych do aktualnie istniejących technologii; obecnie badania koncentrują się na sieciach korzystających z → protokołów komunikacyjnych → TCP/IP, stanowiących podstawę → Internetu, oraz na wykorzystaniu bardzo dużej przepustowości istniejących połączeń światłowodowych, pozwalającej na ujednoczenie sieci do transmisji danych, głosu, obrazu i innych informacji multimedialnych. Są badane i udoskonalane najpopularniejsze protokoły, szczególnie → poczty elektronicznej oraz sieci → WWW i WAP; wyniki w zakresie wykorzystania języka → Java oraz technik kompresji danych (np. filmów) pozwalają na przesyłanie przez istniejące sieci coraz bardziej złożonych informacji. Jednocześnie są prowadzone prace nad zapewnieniem bezpieczeństwa informacji, zwłaszcza ze względu na korzystanie z sieci do transakcji gosp. i finansowych; badania obejmują w szczególności sposoby szyfrowania informacji (→ kryptologia).

**Inżynieria oprogramowania** obejmuje badania nad przemysłowym wytwarzaniem oprogramowania i nadzorowaniem cyklu życia programu, począwszy od analizy wymagań, przez projektowanie i wykonywanie programu, po jego wdrożenie, eksploatację i modernizację lub zastąpienie innym programem; celem jest ograniczenie błędów, przyspieszenie procesu tworzenia oprogramowania oraz zapewnienie przewidywalności i powtarzalności tego procesu. Szczególnie istotne jest określenie (modelowanie) sposobów opisywania wymagań użytkowników, zwykle z wykorzystaniem diagramów graficznych oraz narzędzi programowych → CASE. Z opracowanych metod obecnie najpopularniejsze są: strukturalna i obiektowa; w obydwu są modelowane zarówno używane dane, jak i procedury ich wprowadzania, przetwarzania i wyliczania wyników. Na podstawie zdefiniowanych modeli można zaprojektować oprogramowanie (m.in. podział na moduły funkcjonalne, sposób zapisu danych z wykorzystaniem konkretnej technologii baz danych, sposób komunikacji z użytkownikiem z wykorzystaniem wybranego → interfejsu), a często także wspomóc wykonanie prototypu, na podstawie którego można ocenić prawidłowość przyjętych wymagań i decyzji projektowych. Przedmiotem badań są także sposoby organizacji zespołu programistów, metody testowania i dokumentowania. Ze względu na postulowaną powtarzalność są prowadzone prace nad technikami prognozowania czasu wykonania programu we wczesnych fazach prac (np. na podstawie analizy wymagań). Sztuczna inteligencja obejmuje badania w wielu różnych obszarach, dotyczących rozwiązywania problemów, dla których nie można użyć skutecznych algorytmów i trzeba stosować metody przybliżone (heurystyczne), naśladujące sposób rozumowania człowieka lub zwierzęcia; są to np.: rozpoznawanie obrazów (rozpoznawanie obiektów trójwymiarowych i związków między nimi), rozpoznawanie mowy, przetwarzanie języka naturalnego (rozumienie wypowiedzi w języku naturalnym, tłumaczenie między różnymi językami naturalnymi oraz generowanie wypowiedzi w języku naturalnym), wyciąganie wniosków z informacji niepewnej i niepełnej, prowadzenie rozgrywek (np. w szachy). Praktyczne zastosowanie znajdują szczególnie systemy doradcze, pozwalające na zapisanie w sformalizowanej postaci wiedzy ekspertów, a następnie jej masowe wykorzystanie, np. przy wstępnym diagnozowaniu medycznym. Wprawdzie nie jest jeszcze możliwe automatyczne tłumaczenie dowolnych wypowiedzi w języku naturalnym na inny język naturalny, ale można tłumaczyć teksty specjalistyczne, budowane z wykorzystaniem sztywnych reguł (np. opisy patentowe). Istniejące systemy rozpoznawania mowy pozwalają na rozpoznawanie prostych poleceń lub zdań i są stosowane np. do sterowania urządzeniami przez osoby niepełnosprawne, dyktowania prostych listów oraz notatek; trwają prace nad zwiększaniem liczby rozpoznawanych słów oraz wykrywania związków logicznych między nimi. Rozpoznawanie obrazów pozwala na automatyczne wczytywanie tekstów drukowanych lub pisanych ręcznie, a także na automatyczne sterowanie bezzałogowymi urządzeniami (np. rakiety samonaprowadzające na cel, sondy kosm.). Badania nad programami do gry (np. w szachy) spowodowały rozwinięcie teorii podejmowania decyzji w warunkach niepełnej informacji (→ decyzji teoria).

**Historia.** Za pierwsze badania w dziedzinie informatyki można uznać prace Ch. Babbage'a (ok. 1830), który zaprojektował maszynę liczącą (najpierw tzw. maszynę różnicową, potem analityczną, zdolną do wykonywania obliczeń wg zadanego algorytmu). W 1890 w USA wyniki spisu powszechnego zostały opracowane za pomocą maszyn licząco-sortujących H.

Holleritha, wykorzystujących karty perforowane; maszyny te działały na podstawie stałego (niemodyfikowalnego) programu. Prace nad projektowaniem i budową elektromechanicznych, a następnie elektronicznych maszyn cyfrowych, początkowo przeznaczonych do wykonywania obliczeń dla celów wojskowych, a potem także do przetwarzania innych informacji, rozpoczęto w latach 40. XX w.

Wcześniej natomiast, w latach 30. XX w. rozpoczęto badania w dziedzinie logiki, poświęcone definicji algorytmu i określeniu klasy problemów (funkcji), które mogą być rozwiązywane (obliczane) przez te algorytmy (A. Turing, K. Gödel, A. Markow, A. Church, E. Post, S. Kleene). Doprowadziły one do stworzenia wielu definicji (m.in. maszyny Turinga) i wykazania ich równoważności; okazało się, że istnieją problemy, które nie mogą być rozwiązane w skończonym czasie przez żaden algorytm; wyniki dotyczące algorytmów są powiązane z wynikami dotyczącymi funkcji rekurencyjnych oraz nierozstrzygalności arytmetyki.

Powstanie pierwszych komputerów spowodowało konieczność podjęcia badań nad sposobem najlepszego wykorzystania istniejącej technologii oraz tworzenia skutecznych i efektywnych programów. W latach 40. i 50. komputery służyły głównie do obliczeń numerycznych (→ numeryczne metody), na podstawie algorytmów wykorzystywanych wcześniej przy obliczeniach ręcznych i elektromechanicznych (np. algorytm Gaussa rozwiązywania układów równań liniowych). Ze względu na małą moc obliczeniową i konieczność maksymalizacji jej wykorzystania programy były pisane bezpośrednio w językach wewnętrznych (→ języki programowania) poszczególnych komputerów. Przewidywania przyszłości komputerów doprowadziły do teoretycznych badań w zakresie sztucznej inteligencji, m.in. 1950 C.E. Shannon opisał podstawy programu grającego w szachy, a Turing sformułował tzw. test Turinga, określający warunki uznania systemu komputerowego za urządzenie inteligentne.

**Informatyka jako samodzielny dział nauki** wyodrębniła się ok. 1960, wraz z rozwojem zastosowań komputerów, rozbudową baz danych, powstawaniem języków programowania i systemów operacyjnych. Konieczność formułowania algorytmów w postaci niezależnej od komputera, który je wykonuje, a równocześnie dążenie do zwiększenia wydajności programisty doprowadziły do powstania pierwszych języków programowania: 1957 opracowano pierwszą wersję języka → Fortran do obliczeń numerycznych, 1959 → COBOL do zastosowań gospodarczych i finansowych, 1960 → LISP — do przetwarzania informacji zapisanych w postaci symbolicznej, 1962 APL — do obliczeń numerycznych w trybie interakcyjnym. Zdobyte doświadczenia pozwoliły 1960 opublikować opis języka → Algol 60, po raz pierwszy z wykorzystaniem gramatyki formalnej do zdefiniowania składni; Algol 60 miał być językiem wzorcowym, używanym do formułowania i publikowania algorytmów; mechanizmy wówczas wprowadzone (zasady widoczności zmiennych, definiowanie funkcji i procedur, przekazywanie wartości parametrów) są używane do dziś. Dalszy rozwój teorii języków programowania doprowadził do powstania 1968 języka Algol 68, a 1967 — języka Simula 67, w którym wprowadzono koncepcję klas i dziedziczenia, stanowiącą podstawę programowania obiektowego. Równocześnie (na pocz. lat 60.) powstały pierwsze systemy operacyjne, ściśle związane z konkretną konfiguracją systemu komputerowego, a często także z używanym językiem programowania (np. FMS — Fortran Monitor System firmy IBM); systemy te działały w trybie → wsadowym; wraz z wprowadzeniem rodziny komputerów IBM/360 został zbudowany system operacyjny OS/360; jego wielkość i złożoność spowodowały ogromne problemy projektowe i programistyczne, w wyniku których rozpoczęto systematyczne prace w dziedzinie inżynierii oprogramowania. W poł. lat 60. podjęto prace nad systemem operacyjnym Multics, pozwalającym pracować w trybie → interakcyjnym, a równocześnie sformułowano teoretyczne podstawy współbieżności, opisujące zachowanie się takich systemów (E.W. Dijkstra, P. Brinch Hansen). W 1969 w USA uruchomiono sieć → Arpanet, łączącą ośrodki akademickie i badawcze; początkowym celem było zdalne udostępnianie zasobów komputerowych, pozwalające na ich pełniejsze wykorzystanie (zwłaszcza po uwzględnieniu różnicy czasu na obszarze USA); w praktyce sieć była używana przede wszystkim do wymiany informacji (w tym przez pocztę elektroniczną). W 1962 C.A. Petri zaproponował opis działania systemów rozproszonych z wykorzystaniem sieci. W 1966 J. Weizenbaum stworzył (w języku SNOBOL) program



ELIZA, demonstrujący, jak przy użyciu bardzo ograniczonych mechanizmów można stworzyć wrażenie programu rozumiejącego tekst w języku naturalnym.

W latach 60. poczyniono także znaczne postępy w dziedzinie tworzenia efektywnych algorytmów. W 1962 C.A.R. Hoare opisał algorytm tzw. szybkiego sortowania przez podział (ulepszony 1969 przez R.C. Singletona). Sformułowano wiele algorytmów związanych z teorią grafów, m.in. 1959 Dijkstra podał optymalny algorytm wyznaczania najkrótszych dróg o ustalonym początku. W 1965 J.M. Cooley i J.W. Tukey opisali algorytm FFT (szybka transformacja Fouriera). W 1969 V. Strassen opisał algorytm szybkiego mnożenia macierzy; badania w tej dziedzinie podsumował D. Knuth (1968–73). W 1968 amerykańska organizacja zawodowa ACM opracowała pierwszy ramowy program studiów informatycznych. Na przełomie lat 60. i 70. rozpoczęto badania w zakresie analizy poprawności programów; 1967 R.W. Floyd zaproponował sposób dowodzenia własności programów zapisanych w postaci schematów blokowych; Hoare i Dijkstra zastosowali ten sposób do analizy programów zapisanych w językach programowania, a 1968 Dijkstra wprowadził pojęcie programowania strukturalnego, bez używania instrukcji skoku. Zgodnie z tymi postulatami 1971 N. Wirth zaprojektował język → Pascal. W 1969 Z. Manna wprowadził pojęcia częściowej i całkowitej poprawności. Rozszerzono stosowanie metod mat.; 1971 D.S. Scott i C. Strachey zaproponowali opis programu jako funkcji w pewnej algebrze, zdefiniowanej przez zbiór równań odpowiadających kolejnym instrukcjom (semantyka denotacyjna). Po wejściu do powszechnego użycia pamięci dyskowych, 1970, E.F. Codd zaproponował zastosowanie algebry relacji do opisu działania systemów baz danych, tworząc podstawy relacyjnego modelu baz danych oraz języka SQL. Poszukując sposobu zwiększenia mocy języków programowania, 1970 A. Colmerauer zdefiniował język Prolog, w którym program składa się z ciągu reguł wnioskowania w pewnym systemie logicznym.

Powstanie minikomputerów spowodowało rozszerzenie zastosowań komputerów na nowe dziedziny, m.in. przygotowywanie publikacji. Dla ułatwienia korzystania z systemu komputerowego D.M. Ritchie i K. Thompson 1969 stworzyli prosty system operacyjny → Unix dla minikomputera PDP-9, przeniesiony następnie na minikomputer PDP-11, a 1973 zapisany w nowopowstałym języku programowania → C (*inform.*). Wiele wprowadzonych wówczas mechanizmów stało się częścią wszystkich późniejszych systemów operacyjnych. W latach 70. rozwinięto badania problemów, dla których nie znaleziono algorytmów o złożoności wielomianowej; wprowadzono pojęcie problemów NP-pełnych (S.A. Cook 1971, R.M. Karp 1972). Badania nad systemami operacyjnymi doprowadziły do sformułowania teorii procesów współbieżnych i opisu sposobów ich komunikacji, m.in. Dijkstra 1971 opublikował tzw. problem pięciu filozofów, poglądowo obrazujący zagadnienia z tej dziedziny, i zdefiniował tzw. semaforey; 1974 Hoare zaproponował synchronizowanie procesów za pomocą tzw. monitorów, a 1978 przedstawił propozycję notacji CSP (ang. Communicating Sequential Processes), która została zastosowana do synchronizacji procesów m.in. w języku → Ada.

Rozwinięto badania nad kryptografią; 1976 W. Diffie i M. Hellman zaproponowali zastosowanie klucza → publicznego do szyfrowania przesyłanych danych, a 1978 R.L. Rivest, A. Shamir i L. Adelman zaproponowali pierwszy algorytm korzystający z takiego klucza (od pierwszych liter nazwisk twórców zw. RSA).

Wraz ze wzrostem wielkości oprogramowania rozpoczęto badania nad procesem jego tworzenia. W 1979 E. Yourdon zaproponował zastosowanie diagramów graficznych do opisu wymagań funkcjonalnych (obecnie stanowią one podstawę analizy strukturalnej). W 1979 program komputerowy H. Berlinera wygrał mecz z mistrzem świata w trik-traka. W 1979 firma Xerox na własne potrzeby opracowała rodzinę mikrokomputerów Alto, korzystających z → graficznego interfejsu użytkownika, połączonych siecią lokalną → Ethernet. Popularyzację tych rozwiązań przyniosło masowe rozpowszechnienie komputerów osobistych w latach 80.; pojawiły się wówczas nowe typy programów, działających w trybie interakcyjnym: → arkusz kalkulacyjny, → edytor tekstu. Wraz z zastosowaniem mikroprocesorów nastąpiło zmniejszenie liczby różnych architektur komputerów oraz systemów operacyjnych; wprowadzenie efektywnych → kompilatorów spowodowało wzrost popularności języków programowania wyższego poziomu, przede

wszystkim języka C (do lat 70. ok. 50% programów było pisanych w → assemblerach, a ok. 45% w COBOLU). Pojawienie się w latach 90. narzędzi analitycznych i projektowych klasy CASE, narzędzi programowania klasy RAD (ang. Rapid Application Development 'szybkie tworzenie programów aplikacyjnych') oraz systemów zarządzania relacyjnymi bazami danych, korzystających z języka SQL, pozwoliło znacznie zwiększyć wydajność pracy przy tworzeniu oprogramowania i podnieść jego jakość.

W 1983 w sieci Arpanet wprowadzono protokół TCP/IP, tworząc zaczątek Internetu. Masowy wzrost liczby komputerów oraz wprowadzanie w telekomunikacji łączy cyfrowych spowodowało jego lawinowy rozwój, szczególnie po zdefiniowaniu 1991 przez T. Berners-Lee języka → HTML i protokołu HTTP, używanych do dostępu do stron WWW. Wzrost mocy obliczeniowej komputerów spowodował rozwój nowych zastosowań, m.in. w → grafice komputerowej (produkcja filmów, kolorowanie filmów czarno-białych), symulacji zjawisk przyr. (cyfrowe tunele aerodynamiczne, symulacja wybuchów jądrowych, prognozowanie pogody), a także sztucznej inteligencji (1998 program komputerowy po raz pierwszy wygrał mecz szachowy z mistrzem świata). Obecnie głównymi ośrodkami badawczymi w dziedzinie informatyki są: wyższe uczelnie (m.in. w USA — Massachusetts Institute of Technology, uniwersytety Stanforda i Carnegie-Mellon, w Europie — uniwersytet w Oxfordzie, politechnika w Zurychu), jednostki badawcze finansowane przez rządy (niem. GMD Forschungszentrum Informationstechnik GmbH, fr. Institut National de Recherche en Informatique et en Automatique), laboratoria i instytuty przem. prowadzone przez firmy inform. i komputerowe (np. IBM — Thomas J. Watson Research Center, Xerox — Palo Alto Research Center, DEC, a później Compaq — Systems Research Center, Bell — Bell Labs).

**Informatyka w Polsce.** Tak jak wszędzie, początkowo informatyka teoretyczna była rozwijana w ścisłym związku z pracami nad budową sprzętu. Pierwszą jednostką badawczą z tej dziedziny była Grupa Aparatów Matematycznych, powstała 1948 w Państwowym Instytucie Matematycznym w Warszawie. W 1956 powstał Ośrodek Obliczeniowy Instytutu Badań Jądrowych (obecnie Instytut Podstaw Informatyki PAN). Na Uniwersytecie Warszawskim 1964 powołano Zakład Obliczeń Numerycznych; równocześnie powstała Katedra Metod Numerycznych (obecnie Instytut Informatyki). W 1973 na Uniwersytecie Warszawskim rozpoczęto pierwsze w Polsce magisterskie studia informatyczne. We Wrocławiu rozwój badań informatycznych był stymulowany przez powstanie w 1959 Wrocławskich Zakładów Elektronicznych Elwro — późniejszego producenta komputerów Odra i Riad. Na Uniwersytecie Wrocławskim opracowano m.in. assembler Most 1 dla komputera Odra 1003, kompilator języka Algol 60 dla komputera Odra 1204, system operacyjny MASON i bibliotekę ok. 200 algorytmów numerycznych. Obecnie prace badawcze w dziedzinie informatyki są prowadzone m.in. w Instytucie Podstaw Informatyki oraz Instytucie Informatyki Teoretycznej i Stosowanej PAN, na uniwersytetach: Warszawskim, Wrocławskim, Gdańskim i UAM w Poznaniu; na politechnikach: Poznańskiej, Warszawskiej, Wrocławskiej, Śląskiej i Lubelskiej, a także na AGH, w Akademii Ekonomicznej w Poznaniu i WAT.

A.V. Aho, J.E. Hopcroft, J.D. Ullman *Projektowanie i analiza algorytmów komputerowych*, Warszawa 1983; D.Harel *Rzecz o istocie informatyki – algorytmika*, Warszawa 1992; R. Ligonnière *Prehistoria i historia komputerów*, Wrocław 1992; J. Madey, M.M. Sysło *Początki Informatyki w Polsce*, "Informatyka" nr 9–10 2000; P. Brinch Hansen (red.) *Classic Operating Systems: from batch processing to distributed systems*, Springer-Verlag 2001; D. Knuth *Sztuka programowania*, 2001.

Jarosław Deminet