

**Po co matematyka
dla informatyka ?**

Odpowiedź, której można udzielić na ten moment brzmi...

... to zależy 😊

Na jakim poziomie powinienem znać matematykę ?

Jeżeli planujesz zajmować się:

- **projektowaniem interfejsów graficznych,**
- **tworzeniem prostych stron internetowych,**
- **administracją sieci komputerowej**

to w większości tych przypadków podstawowa znajomość matematyki (arytmetyka, podstawy logiki, geometria) wystarczy.

Na jakim poziomie powinienem znać matematykę ?

Jeżeli jednak bardziej interesuje Cię:

- **tworzenie gier video,**
- **bycie wszechstronnym programistą,**
- **sztuczna inteligencja,**
- **robotyka,**
- **kryptografia i kryptologia,**
- **big data**
- **praca naukowa**

to Twoja znajomość matematyki powinna wykraczać poza to, co można nazwać podstawami.

Trochę historii

Sama definicja informatyki mówi już o tym w jaki sposób jest ona połączona z matematyką:

Informatyka – dyscyplina nauki zaliczana do **nauk ścisłych** oraz techniki zajmująca się **przetwarzaniem informacji**, w tym również technologiami przetwarzania informacji oraz **technologiami wytwarzania systemów** przetwarzających informacje. **Początkowo stanowiła część matematyki**, później rozwinęła się do odrębnej dyscypliny – pozostaje jednak nadal w ścisłej relacji z matematyką, która **dostarcza informatyce podstaw teoretycznych**.

Trochę historii

Jako konkluzję definicji informatyki warto przytoczyć tutaj słowa Profesora Dijkstry:

„Informatyka ma tyle samo wspólnego z komputerami, co astronomia ma z teleskopami.”

Trochę historii

Jedną z ikon kryptografii jest maszyna szyfrująca Enigma, która została zaprojektowana na początku lat 20 XX wieku. Najszerze zastosowanie znalazła podczas Drugiej Wojny Światowej do szyfrowania wiadomości wymienianych między sztabami wojsk niemieckich.

Do jej złamania znacząco przyczynili się Polacy w osobach Mariana Rejewskiego, Jerzego Różyckiego i Henryka Zygalskiego z Biura Szyfrów Oddziału II Sztabu Głównego Wojska Polskiego co nastąpiło w 1932 roku. Enigma była wciąż udoskonalana co zwiększało ilość możliwych kluczy szyfrowania, dlatego prace nad łamaniem tych szyfrów były kontynuowane również we Francji i Wielkiej Brytanii już po wybuchu wojny.

Poza Polską najczęściej słyszy się nazwisko Alana Turinga, który wraz ze swoim zespołem rozpoczął prace nad budową deszyfratora Enigmy bazując na materiałach polskich naukowców.



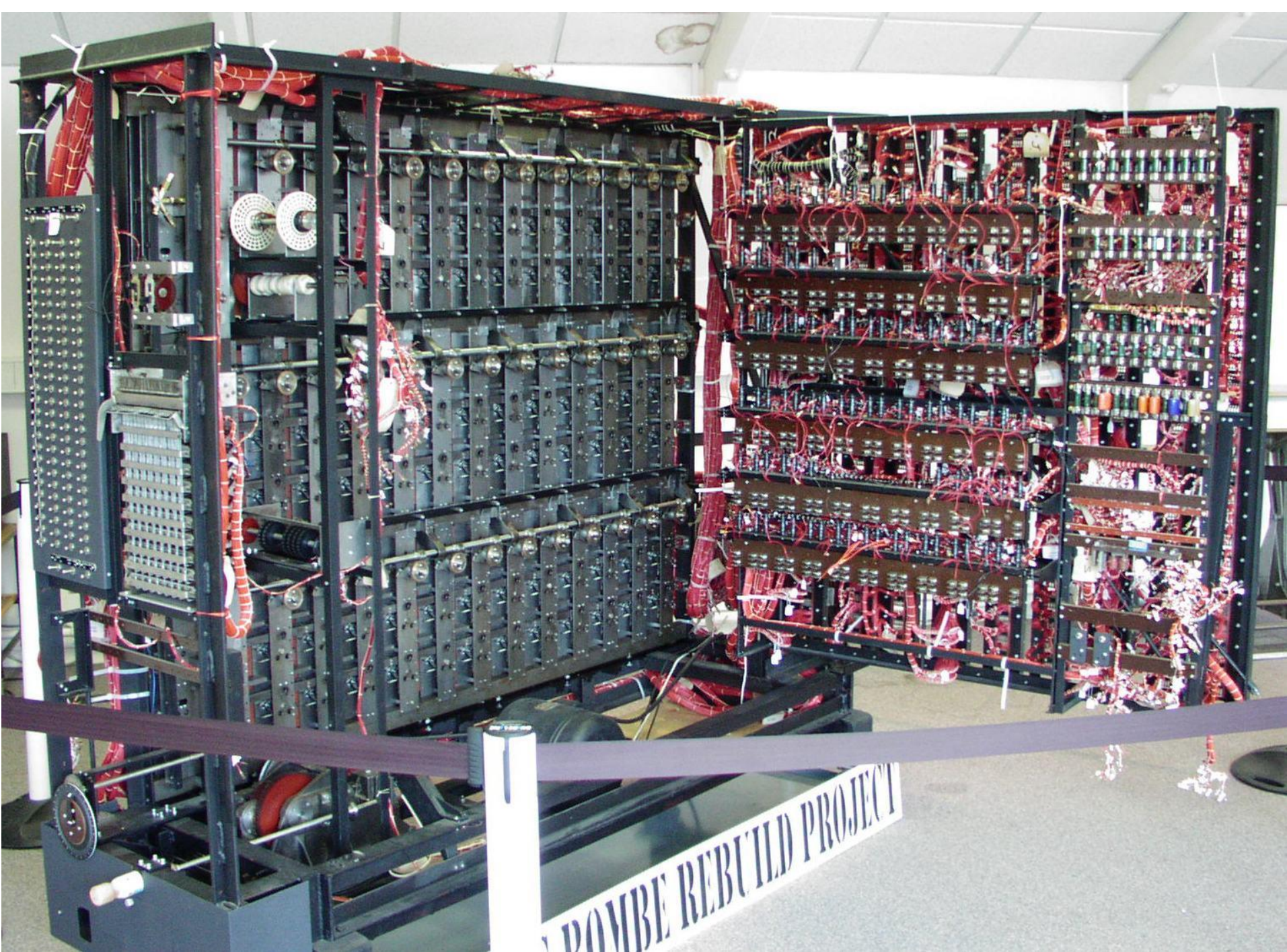
Źródło: wikipedia

Trochę historii

Praca Turinga przy Enigmie dała początek wielu rozwiązaniom stosowanym dziś w informatyce. Przytoczę tutaj fragment z jego biografii:

„Przebywając w Cambridge Turing napisał swoją prawdopodobnie najważniejszą pracę matematyczną **On Computable Numbers**, czyli O liczbach obliczalnych. To właśnie w niej wprowadził abstrakcyjną maszynę, która była w stanie **wykonywać zaprogramowaną matematyczną operację, czyli algorytm**. Maszyna mogła wykonać tylko jeden, określony algorytm, na przykład mogła podnieść liczbę do kwadratu, podzielić, dodać, odjąć. Według Turinga liczby miały być podawane maszynie za **pomocą papierowej taśmy podobnej do taśmy z melodią zapisaną dla pianoli**. W swojej pracy Turing opisał wiele takich maszyn, które uzyskały wspólne miano **maszyn Turinga**.

Po wojnie zaprojektował jeden z pierwszych elektronicznych, programowanych komputerów. Był również pomysłodawcą tak zwanego testu Turinga – eksperymentu będącego próbą formalnego zdefiniowania sztucznej inteligencji. ”



BOMB REBUILD PROJECT

Trochę historii

Test Turinga – sposób określania zdolności maszyny do posługiwania się językiem naturalnym i pośrednio mającym dowodzić opanowania przez nią umiejętności myślenia w sposób podobny do ludzkiego. W 1950 roku Alan Turing zaproponował ten test w ramach badań nad stworzeniem sztucznej inteligencji – zamiast pełnego emocji i w jego pojęciu bezsensownego pytania "Czy maszyny myślą?" na pytanie lepiej zdefiniowane.

We wrześniu 2011 roku program CleverBot, stworzony przez Rollo Carpentera w 1988 roku i od tego czasu rozwijany, oszukał ponad 59,3% rozmówców, myślących, że jest on człowiekiem. Do zaliczenia testu Turinga zabrakło 4%, gdyż człowieka na tych samych zawodach rozpoznało prawidłowo 63,3% osób.

Matematyka i AI

Sztuczna inteligencja:

- dynamiczny rozwój (trend),
- potrzeba nowych algorytmów, formuł matematycznych,
- ograniczone możliwości sprzętu,
- lawinowo rosnąca ilość dziedzin, gdzie AI znajduje zastosowanie.

Matematyka i AI

Przykłady zastosowania sztucznej inteligencji:

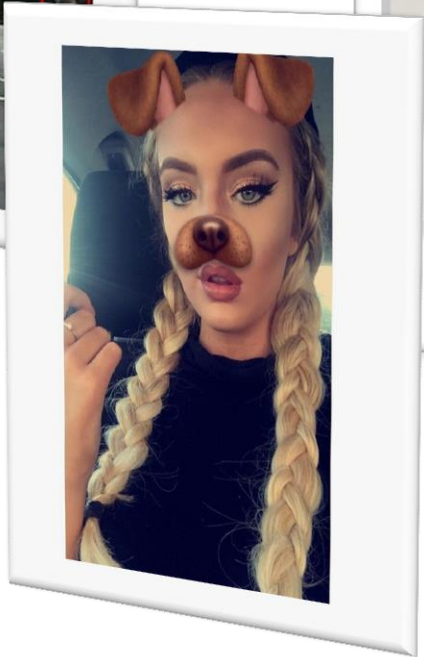
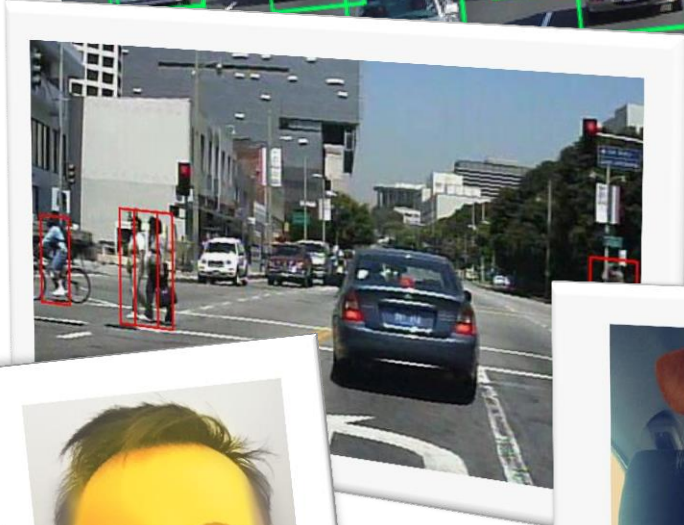
- przetwarzanie obrazów,
- rozpoznawanie mowy,
- NLP (natural language processing),
- gry takie jak szachy, go, sztuczna inteligencja w grach video,
- wnioskowanie i podejmowanie decyzji w przypadku braku wszystkich danych.

Matematyka i AI

Matematyka i przetwarzanie obrazów

Obraz (zdjęcie) jest zbiorem informacji (**pikseli**), które stanowią całość. Obraz ma swoje rozmiary (**rozdzielczość**), która określa ilość pikseli w pionie i poziomie a ich iloczyn to liczba pikseli danego obrazu. Mamy więc do czynienia z **macierzą** pikseli. Wiele operacji, które wykonywane są na obrazach za pomocą programów graficznych (np. filtry) to operacje na macierzach właśnie.

Inne bardziej zaawansowane funkcje (redukcja efektu czerwonych oczu, zaznaczanie obiektów, tła) często bazują na algorytmach, które należą do dziedziny sztucznej inteligencji. Bardzo szerokie zastosowanie mają tutaj sieci neuronowe, gdzie informacje o obrazie zapisywane są w postaci wektora lub macierzy pikseli, aby usprawnić obliczenia.



Matematyka i AI

Inne praktyczne zastosowania przetwarzania obrazów:

- zakupy,
- szukanie podobnych obrazów,
- fotografia,
- systemy na lotniskach,
- autonomiczne samochody, drony,
- captcha, OCR
- Google maps
- wiele, wiele innych...

Matematyka i AI

Zagadnienia matematyczne wykorzystywane w CV (Computer Vision):

- wektory, macierze,
- geometria,
- trygonometria,
- złożoność obliczeniowa (big O notation),
- statystyka,
- rachunek prawdopodobieństwa,
- wnioskowanie Bayesowskie,
- algebra liniowa,
- przyszłość ???

Matematyka i AI

Rozpoznawanie mowy, synteza mowy, Natural Language Processing.

Algorytmy rozpoznawania mowy to kolejny „efektowny” sposób na zaprezentowanie możliwości jakie oferuje sztuczna inteligencja i zaadoptowane matematyczne formuły.

Po krótkce można opisać powyższe zagadnienia jako:

rozpoznawanie mowy --> tekst
tekst ---> mowa syntetyczna

Natomiast zagadnienia **przetwarzania języka naturalnego** (NLP) polegają na badaniu konstrukcji języka, analizowaniu budowy zdań po to, aby rozumieć znaczenie danego zwrotu, zdania i na podstawie tych informacji móc wykonać stosowną operację. Ta część sztucznej inteligencji łączy również inne dziedziny oprócz matematyki i informatyki np. lingwistykę.

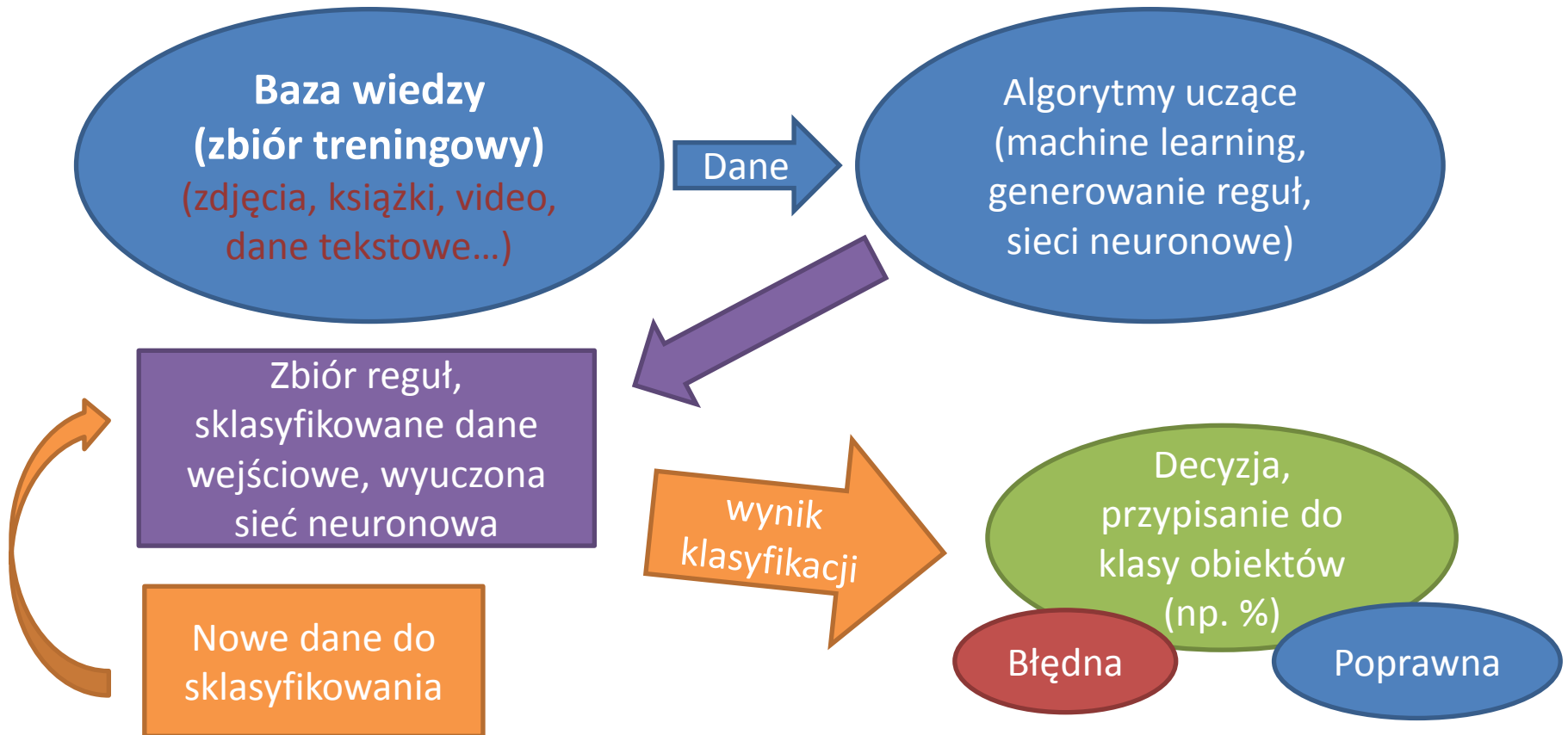
Matematyka i AI

Przykładowe zastosowania:

- medycyna – transkrypcja diagnozy, czynności wykonywanych podczas operacji,
- filmy video – tłumaczenie mowy na tekst w czasie rzeczywistym, lektor,
- edytory tekstu,
- Google Now, Apple Siri,
- komunikacja miejska,
- tłumaczenie rozmów telefonicznych czy video konferencji w czasie rzeczywistym,
- czatboty,
- systemy eksperckie, gdzie zadajemy pytanie w formie tekstu (lub mowy) i otrzymujemy odpowiedź zgodną z tematem pytania (NLP),
- opowiadania i krótkie powieści pisane przez sztuczną inteligencję,
- systemy antyspamowe, antywirusowe.

Matematyka i AI

Jak działają algorytmy wizji komputerowej (CV) oraz przetwarzania mowy/tekstu ?



Matematyka i AI

How many shortest-length paths are there to get from your house to the doughnut shop?



$$\binom{n}{k} = \frac{n!}{(n-k)!k!}$$

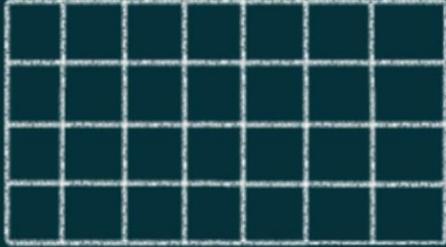
$$e^{i\pi} + 1 = 0$$

P	Q	R	P ∨ Q	P ∨ R	(P ∨ Q) ∧ (P ∨ R)
T	T	T	T	T	T
T	T	F	T	T	T
T	F	T	T	T	T
T	F	F	T	T	T
F	T	T	T	T	T
F	T	F	T	F	F
F	F	T	F	T	F
F	F	F	F	F	F

7, 11, 15, 19, 23...

$$\begin{aligned} a_1 - a_0 &= 4 \\ a_2 - a_1 &= 4 \\ a_3 - a_2 &= 4 \\ &\vdots \\ + a_n - a_{n-1} &= 4 \\ \hline a_n - a_0 &= 4n \\ a_n &= a_0 + 4n \end{aligned}$$

4 up's
↑
7 right's
→



1 0 1

1 0 1 B₇

1 0 1 0 0 0 1 0 1

B₄



One-to-One

Find 7 + 12 + 17 + 22 + ... + 342.

$$S_n = 7 + 12 + 17 + 22 + \dots + 342$$

$$+ S_n = 342 + 337 + 332 + 327 + \dots + 7$$

$$2S_n = 349 + 349 + 349 + 349 + \dots + 349$$

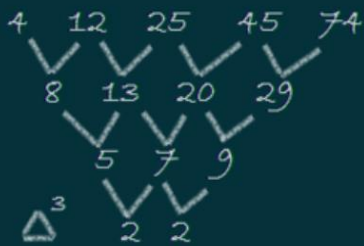
$$2S_n = 349 \cdot 68$$

$$S_n = \frac{349 \cdot 68}{2}$$

$$S_n = 11866$$



$$\binom{11}{7} = \binom{11}{4} = 330 \text{ paths}$$



Onto



There are six dogs to give 13 tacos.

Use a 'stars and bars' diagram to illustrate the first and sixth dog get 3 tacos, the second dog gets none, the third dog gets 5 and the fourth dog gets one.



$$A = \{2, 4, 10, \text{doughnut}\}$$

$$(A \cup B \cup C) \cup (A \cap B \cap C)$$



$$v - e + f = 2$$

P.I.E. Example:

$$6! - \left[\binom{6}{1}5! - \binom{6}{2}4! + \binom{6}{3}3! - \binom{6}{4}2! + \binom{6}{5} - 1 \right]$$

Original:
 $\exists x \forall y (x \geq 2y \rightarrow x > y + 1)$

Converse:
 $\exists x \forall y (x > y + 1 \rightarrow x \geq 2y)$

Negation:
 $\neg [\exists x \forall y (\neg(x \geq 2y) \vee x > y + 1)]$
 $\forall x \exists y (x \geq 2y \wedge x \leq y + 1)$

Contrapositive:
 $\exists x \forall y (x \leq y + 1 \rightarrow x < 2y)$

Matematyka i AI

Więc co ze świata matematyki jest wykorzystywane w sztucznej inteligencji ?

Oto kilka zagadnień:

- regresja,
- teoria zbiorów,
- teoria informacji,
- teoria grafów,
- rachunek różniczkowy,
- teoria mnogości,
- pojęcie optymalizacji,
- matematyka dyskretna...

Chcę być programistą !

Czy dobra znajomość matematyki jest niezbędna aby zostać programistą ?

Nie.

Czy dobra znajomość matematyki gwarantuje bycie dobrym programistą ?

Nie.

Czy mogę w takim razie założyć, że matematyka nie przyda mi się w pracy programisty ?

Nie.

Chcę być programistą !

Powody, dla których takie założenie jest niebezpieczne to przede wszystkim **brak możliwości przewidzenia** tego co stanie się w branży IT za kilka, kilkanaście lat. Można jednak z pewnym prawdopodobieństwem stwierdzić, że będziemy mieć do czynienia z **dalszym rozwojem sztucznej inteligencji i coraz większą ilością danych** do przeanalizowania (Big Data). Zapotrzebowanie na programistów, którzy będą przygotowani do pracy z tymi zagadnieniami będzie wysokie.

Nawet jeżeli te zagadnienia nie leżą w kręgu zainteresowań przyszłego programisty nie da się wykluczyć, że matematyka się nie przyda. A branża finansowa ? Tworzenie i programowanie urządzeń ? Kryptografia ? Tworzenie gier komputerowych ?

W idealnych warunkach moglibyśmy wybierać, określać przy jakich projektach chcemy i będziemy pracować, ale czy możemy zaryzykować i założyć, że będziemy mieć aż tyle szczęścia ?

Chcę być programistą !

Programiści są mocno narażeni na zjawisko **wypalenia zawodowego**. Jedną z jego przyczyn jest znudzenie aktualną pracą co może się pojawić w przypadku mało wymagających i powtarzających się tematów realizowanych projektów. Dlaczego matematyka mogłaby ograniczyć możliwość wystąpienia takiego efektu ? Jej znajomość podnosi umiejętność rozwiązywania wielu klas problemów (matematyka to również sposób myślenia) oraz daje większe możliwości angażu w innowacyjnych projektach.

Nie ma co ukrywać, że matematyka **NIE ZASZKODZI** żadnemu programiście, a na pewno może pomóc w znalezieniu lepszej pracy zarówno pod kątem nowych wyzwań jak i zarobków.

Chcę być programistą !

A co z grami komputerowymi ?

Dobra wiadomość jest taka, że aby grać najczęściej znajomość matematyki nie jest potrzebna 😊

Jednak aby zająć się na poważnie tworzeniem gier (na własny rachunek czy też pracując dla studia) bez znajomości matematyki, algorytmiki i/lub fizyki ani rusz 😞

Gamedev czyli tworzenie gier

A silniki graficzne ?

Owszem, posiadają wbudowaną bardzo dużą ilość algorytmów odpowiedzialnych za symulacje ruchu, swobodnego spadania, trajektorii, wykrywania kolizji, rozpraszania światła, perspektywy i wiele innych. Jednak funkcje, których dostarczają (API) deweloperom wymagają zdefiniowania wartości poszczególnych parametrów zjawiska, np. generowania fal wodnych, prędkości wylotowej pocisku a wtedy znajomość zasad działania takiego algorytmu jest przydatna lub wręcz niezbędna.

Gamedev czyli tworzenie gier

Matematyka w grach video.

Jednym z zagadnień, których rozwiązanie zostało zaczerpnięte z matematyki jest pojęcie **pathfindingu** czyli szukania optymalnej ścieżki, jaką musi pokonać obiekt (np. postać) aby przemieścić się z aktualnej pozycji do zadanej przez gracza. Można tutaj przytoczyć sławnego pioniera informatyki (bardziej teoretycznej) oraz matematyka **Edsgera Dijkstrę**. Znany jest głównie z **algorytmu odnajdowania najkrótszych ścieżek w grafie**, który nazwany został jego nazwiskiem. Profesor Dijkstra miał również bardzo duży wkład w rozwój języków programowania za co został uhonorowany nagrodą Turinga w roku 1972.

Algorytm ten jest wykorzystywany nie tylko w grach, ale również w algorytmach routingu czy aplikacjach związanych z nawigacją.

Od momentu pojawienia się oryginalnej wersji algorytmu pojawiły się jego zmodyfikowane wersje oraz nowe rozwiązania problemu najkrótszych ścieżek.

Gamedev czyli tworzenie gier

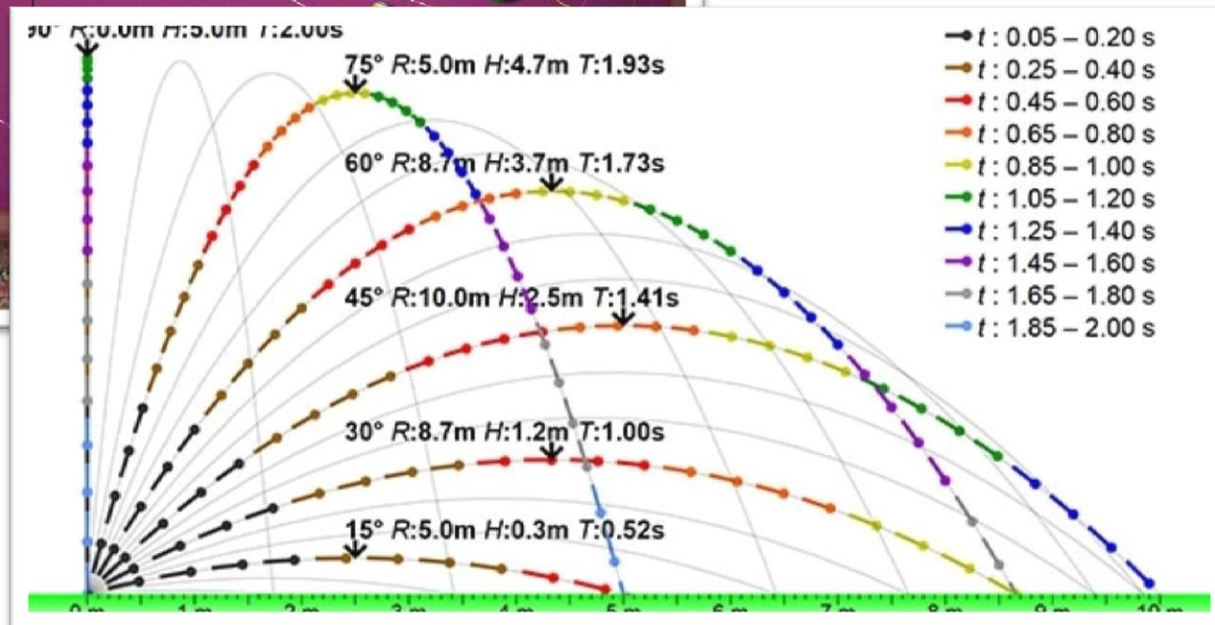
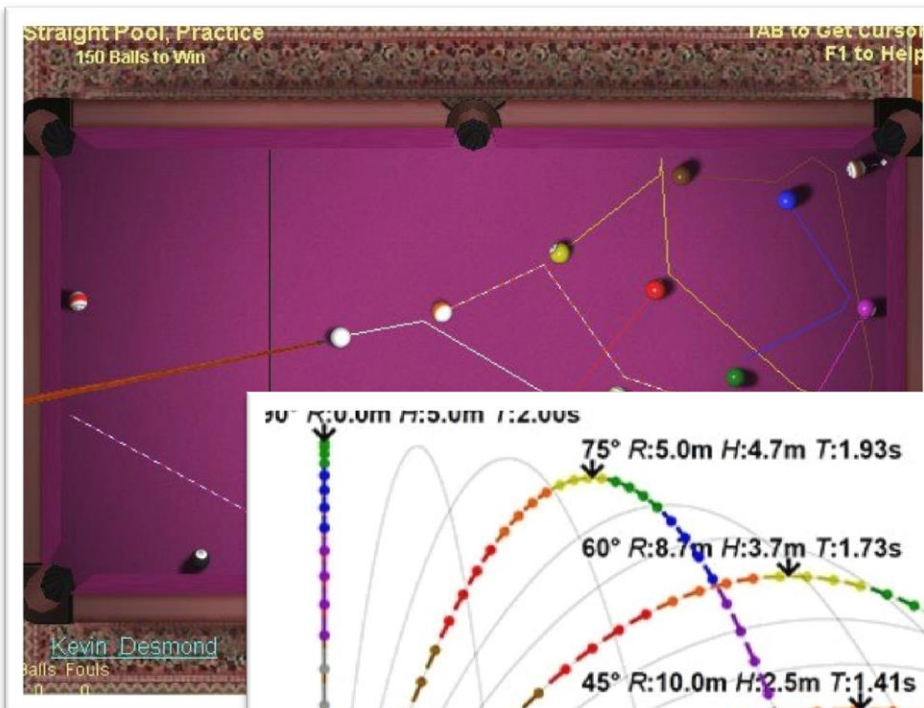
Wielu graczy na pewno zachwyca się bardzo realistycznym odwzorowaniem falującej wody, ale niewielu zdaje sobie sprawę, że do ich wygenerowania wykorzystywane są rachunek całkowy oraz rachunek różniczkowy.

Szczególnie duże znaczenie ma matematyka przy symulowaniu zjawisk związanych z kinematyką, czyli ruchem. W prostych grach (np. 2D) zastosowanie mają m.in. Funkcje kwadratowe, trygonometria, wektory. Ważna jest też znajomość fizyki.

Pojęcie złożoności obliczeniowej jest również ważne w grach i pozwala oszacować jak dana funkcja będzie wpływała na obciążenie procesora co przekłada się na szybkość działania samej gry, a jak wiemy FPS ma znaczenie 😊

Gamedev czyli tworzenie gier

Również **teoria gier** w połączeniu z rachunkiem prawdopodobieństwa oraz statystyką mają szerokie zastosowanie w balansowaniu rozgrywki co jest coraz bardziej istotne w tej dziedzinie ze względu na rosnącą ilość gier on-line. Niewłaściwe zrównoważenie np. postaci może powodować, że będzie ona częściej wybierana od innych, gdyż statystycznie częściej wygrywa. W zbalansowaniu gry mogą również pomóc algorytmy sztucznej inteligencji, które nauczone zasad gry mogą toczyć pojedynki między sobą i dokonywać analizy, która pozwoli ocenić, gdzie należy wprowadzić zmiany.



$y_2 = h = s \sin \theta$
 $0 = K_2 + U_2 - U_1$
 $0 = 1/2 m v_2^2 + mgs - 1/2 ks^2$
 $v_2 = \sqrt{[(ks^2)/m] - 2gs}$

$0 = (K_2 - K_1) + (U_{2z} - U_{1z}) + (U_{2x} - U_{1x})$
 $0 = K_2 + U_2 - U_1$
 $0 = 1/2 m v_2^2 + mgs - 1/2 ks^2$
 $v_2 = \sqrt{[(ks^2)/m] - 2gs}$

$a = 9.8 \text{ m/s}^2$

Gamedev czyli tworzenie gier

Wniosek

Jako programista (deweloper), który planuje zajmować się zagadnieniami sztucznej inteligencji, Big Data, kryptografią lub tworzeniem gier komputerowych niezbędne jest poznanie matematyki na poziomie akademickim, a przynajmniej poszerzenie swoich horyzontów o wiedzę, którą w trakcie studiów zdobyć można. Zdarza się, że przy bardziej złożonych problemach natury programistycznej matematyka przychodzi z pomocą podsuwając pomysły na rozwiązanie, które poznaliśmy na lekcji matematyki.

Znani w IT

Bill Gates

Mimo, że studiów nie ukończył (przerwał je aby założyć Microsoft) był jednym z najlepszych uczniów w szkole średniej (wynik w teście końcowym 1590 na 1600 możliwych punktów przy średniej około 1000).



Znani w IT

Steve Jobs

Podobnie jak Bill Gates studiów nie ukończył (studiował prawo przez rok). Nie osiągał tak dobrych wyników w nauce jak Gates, ale był uznawany za zdolnego ucznia, ale dość dziwnego i sprawiał sporo problemów swoim zachowaniem. Mimo, że nie był geniuszem matematycznym to sukces odniósł, bo „miał swojego Woźniaka”.



Znani w IT

Steve Wozniak

Chociaż został wyrzucony na pierwszym roku studiów, to powód czyli zhakowanie komputera uniwersyteckiego świadczy o nieprzeciętnych umiejętnościach i wiedzy Wozniaka. To on samodzielnie stworzył komputer Apple I, który Jobs umiał sprzedać, co dało początek firmie Apple. Wozniak pracował również dla HP gdzie zajmował się tworzeniem... kalkulatorów.



Znani w IT

Mark Zuckerberg

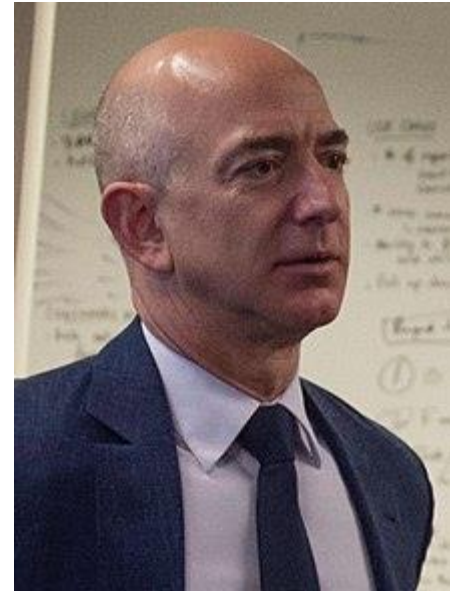
On również (!) nie ukończył studiów w normalnym toku. Dokonał tego dopiero po 12 latach. Zanim jednak w 2004 wspólnie ze znajomymi stworzył Facebooka był autorem odtwarzacza muzyki, czatu, narzędzi do nauki oraz gier. Również fakt, że był studentem Harvardu świadczy o tym, że posiadał odpowiednią wiedzę.



Znani w IT

Jeff Bezos

Założyciel Amazona jako nastolatek często podczas wakacji odwiedzał swojego dziadka, który pracował dla DARPA. To w tej organizacji powstał projekt ARPAnet, który stał się początkiem Internetu. Wykazywał duże zainteresowanie nowymi technologiami. Ukończył studia na University of Florida z wyróżnieniem.



Znani w IT

Sergey Brin

Współtwórca wyszukiwarki Google to syn matematyków rosyjskiego pochodzenia. Po emigracji do USA studiował matematykę i informatykę na Uniwersytecie Stanforda, gdzie poznał Larrego Page'a, z którym wspólnie stworzył wyszukiwarkę Google.



Znani w IT

Linus Torvalds

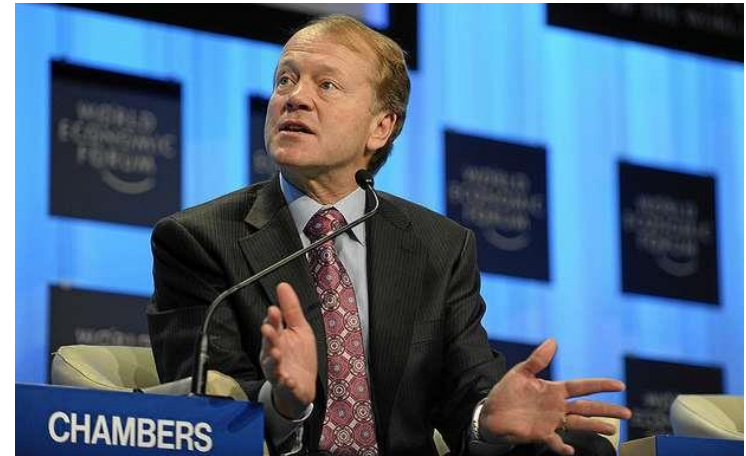
Ten programista pochodzenia fińskiego jest twórcą jądra Linuksa i to między innymi jemu zawdzięczamy rozwój prac nad wolnym oprogramowaniem (Open Source). Programistyczna przygoda Linusa zaczęła się od komputera Commodore, który dostał od dziadka. Wbrew pozorom Linus Torvalds nie skupił się od razu na tworzeniu przełomowych rozwiązań – interesowały go gry komputerowe i sporo czasu poświęcał zarówno na granie, jak i na próby pisania własnych gier.



Znani w IT

John Chambers

To nazwisko zapewne niewielu cokolwiek mówi, ale zapewne firma o nazwie Cisco już tak. Ciekawostką jest to w jaki sposób firma powstała. Otóż John i jego świeżo upieczona żona byli pracownikami Uniwersytetu Stanforda, ale pracowali w oddzielnych jednostkach co utrudniało im komunikację (były takie czasy 😊). Jako, że John był pracownikiem wydziału nauk komputerowych posiadał wiedzę i umiejętności, aby rozpocząć pracę na urządzeniem, które ten problem miało rozwiązać. I tak powstał pierwszy router, który umożliwiał łączenie różnych urządzeń w jedną sieć. Dalsze losy firmy można już podciągnąć pod znany termin „American dream”.



Znani w IT

Maciej Popowicz

Jeden z założycieli portalu nasza-klasa.pl co w niedługim czasie uczyniło go jednym z najmłodszych polskich milionerów. Studiował informatykę w Instytucie Informatyki Uniwersytetu Wrocławskiego. Wielokrotny finalista ogólnopolskich olimpiad informatycznych. Jak sam twierdzi w jednym z wywiadów duża część prac nad projektem wymagała matematycznego podejścia do tworzonych algorytmów serwisu.



Podsumowanie

Mimo, że nie zostały tutaj przedstawione konkretne przykłady wzorów i formuł matematycznych, których używa się w informatyce, to mam nadzieję, że pokazałem co za ich pomocą można osiągnąć.

Jest jeszcze wiele do zrobienia i nic oprócz Nas samych nie stoi na przeszkodzie, abyśmy byli tego częścią.

Podsumowanie

Adresy godne polecenia:

- <https://www.khanacademy.org/>
- <https://www.matemaks.pl/>
- <https://bulldogjob.pl/news/263-znani-programisci-o-programowaniu>

Dziękuję 😊

... i powodzenia !

Źródła

- <https://pl.wikipedia.org/wiki/Enigma>
- https://pl.wikipedia.org/wiki/Alan_Turing#/media/File:Bombe-rebuild.jpg
- <https://www.google.pl/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwi4xqOHpcjYAhVBESwKHSBaDX8QjRwIBw&url=http%3A%2F%2Fwww.trafficvision.com%2F&psig=AOvVaw3gAUCbpwK80oKNI9iJIHak&ust=1515498243613556>
- https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSugMUwiPHjt_pPXASCFnlNn-NFqHkmUmtCNm3iQYcb32XUs7SK
- <https://i.pinimg.com/236x/54/a8/35/54a835a9e239277565a3b14e51cef018--snapchat-filters-puppy.jpg>
- <https://www.extremetech.com/gaming/230252-can-we-finally-admit-that-kinect-is-dead>
- https://cdn-images-1.medium.com/max/2000/1*Jx3X4pD0KdiP8QDhi3d4Og.png
- <http://www.astatix.com/i/pool-3.jpg>
- https://upload.wikimedia.org/wikipedia/commons/thumb/6/61/Ideal_projectile_motion_for_different_angles.svg/350px-Ideal_projectile_motion_for_different_angles.svg.png
- <https://hiciencias.wikispaces.com/file/view/angry%20bird%20tiro%20parabolico.jpg/419944402/344x189/angry%20bird%20tiro%20parabolico.jpg>
- <https://gadzetomania.pl/13572,geniusze-i-wizjonerzy-najwazniejsze-postaci-branzy-it-top-10,all>