

Haladó informatikai algoritmusok

Werner Ágnes

Villamosmérnöki és Információs Rendszerek Tanszék

E-mail: werner.agnes@virt.uni-pannon.hu

„Akinak nincs ideje ma a munkától tanulni, annak a jövőben nem lesz munkája a tudatlanságtól.”

"Ne ránts tőrt, ha egy pofon is megteszi."

Tematika

- Alapfogalmak, Ackermann függvény, a postfix probléma.
- Dinamikus programozás (mátrixok véges sorozatainak szorzása, járdakövezés problémája, nyomtatási feladat, szerelőszalag ütemezési probléma).
- Mohó algoritmusok (esemény-kiválasztási probléma, a mohó stratégia elemei, mohó-választási tulajdonság, optimális részproblémák tulajdonság).
- Ládapakolási algoritmusok (NF, BF, FF, online ládapakolási algoritmusok).
- Mintaillesztés (fogalmak és jelölések, a naív algoritmus, mintaillesztés véges determinisztikus automatával, a Rabin-Karp algoritmus, a Knuth-Morris-Pratt algoritmus, Boyer-Moore algoritmus).
- Megerősítő tanulás (Q-tanulás és alkalmazása).

Segédletek és követelmények

- Honlap: Villamosmérnöki és Információs rendszerek Tanszék <http://www.virt.uni-pannon.hu>
Tantárgyak – Haladó informatikai algoritmusok

Számonkérés:

- Az érdemjegyet a **három dolgozat** eredménye (**70 pont**) és a **beadandó feladat** (**30 pont**) együtt határozza meg. **Minden részfeladatból legalább 40 %-ot kell teljesíteni!**
- pontszám érdemjegy
85-100 jeles (5); 72-84 jó (4); 59-71 közepes (3); 46-58 elégséges (2); 0-45 elégtelen (1)
- **40 ponttól aláírás**
- **Beadandó:** Válassz ki olyan termékeket, szolgáltatásokat, amelyek közel állnak hozzád. Vizsgáld meg milyen módon tudnád ajánlani, pl. egy konkrét felhasználói csoportnak, hogyan lehetne leghatékonyabban elfogadtatni, eladni ezeket másoknak. Próbáld meg kigondolni egy olyan algoritmust, amely ennek a megvalósításában egy alkalmazás részeként a segítségedre lehet. Az alkalmazás teszteléséhez egy nagyon egyszerű változatát az alkalmazásnak készítsd el tetszőleges fejlesztői környezetben.

Leadandó:

1. a megoldandó probléma bemutatása,
2. az algoritmus kódja/pszeudokódja megfelelően kommentálva,
3. az elkészített alkalmazás és annak futtatható változata,
4. egy konkrét lefutás bemutatása.

Csak saját munka fogadható el! Érdemes már most elkezdni vele foglalkozni!

Beadási határidő: 2023. május 14. 24:00

Algoritmus

Definíció: Az **algoritmus** olyan elemi műveletekből kompozíciós szabályok szerint felépített összetett művelet, amelyet megadott feltételt teljesítő bemeneti adataira végrehajtva, a megkívánt kimeneti adatot eredményezi.

Definíció: **Program** = algoritmus leírása adott programozási nyelven.

algoritmusok + adatstruktúrák = programok (Niklaus Wirth)

Az összetett, igazán nagyméretű feladatokkal a **szoftvertchnológia** foglalkozik.

Megnézni: <https://www.youtube.com/watch?v=-DVOeuMECco>
Algoritmusok: a modern élet szabályai

Algoritmusok futási ideje

Legyen A az e_1, \dots, e_m elemi műveletekből felépített algoritmus

Jelölje: t_i az e_i művelet futási idejét

Feltételezzük: $\forall e_i$ elemi művelet futási ideje c_i konstans

Jelölje: $T(A, x)$ az A algoritmus **futási idejét**, ami az x bemeneti adatra ténylegesen végrehajtott elemi műveletek futási idejének az összege

Jelölje: $|x|$ az x bemeneti adat **méretét**

Az e_i elemi műveletek c_i futási ideje és az $|x|$ méret függvény együttesen adja meg a **bonyolultsági mértéket**.

Plussz: memóriaigény, kommunikációs költség

Algoritmusok futási ideje

Legjobb eset: $T_{lj}(A, n) = \min\{T(A, x) : |x| = n\}$

Legrosszabb eset: $T_{lr}(A, n) = \max\{T(A, x) : |x| = n\}$

Átlagos eset: Jelölje $Pr(x)$ annak a valószínűségét, hogy x bemeneti adata lesz az A algoritmusnak.

$$T_a(A, n) = \sum_{|x|=n} Pr(x)T(A, x)$$

Miért tanuljunk algoritmusokat?

Internet. Web search, packet routing, distributed file sharing, ...

Biology. Human genome project, protein folding, ...

Computers. Circuit layout, file system, compilers, ...

Computer graphics. Movies, video games, virtual reality, ...

Security. Cell phones, e-commerce, voting machines, ...

Multimedia. MP3, JPG, DivX, HDTV, face recognition, ...

Social networks. Recommendations, news feeds, advertisements, ...

Physics. N-body simulation, particle collision simulation, ...

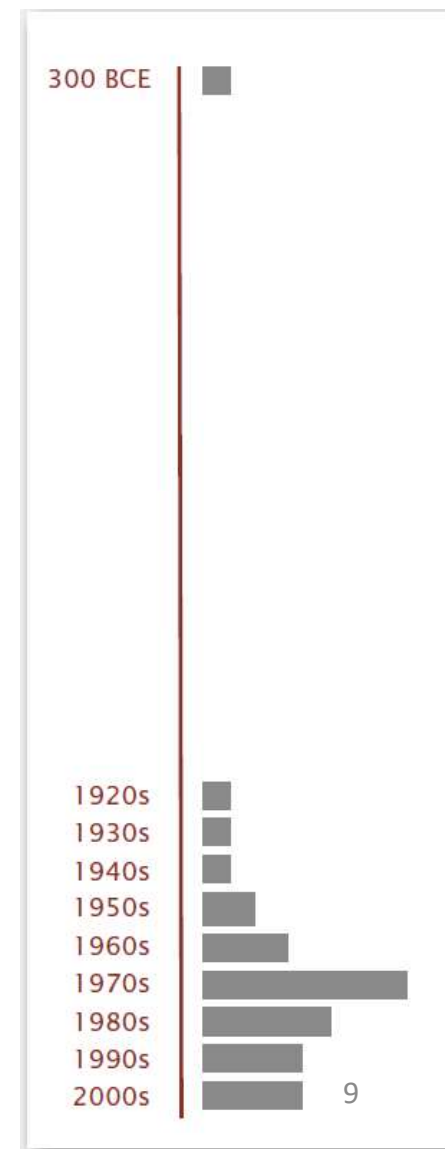
⋮



Miért tanuljunk algoritmusokat?

Régi gyökerek, új lehetőségek

- Az algoritmusok tanulmányozása Euklidész-ig vezethető vissza.
- Church és Turing 1930-ban formalizálta (Church-Turing tézis).
- Néhány fontos algoritmust ilyen kurzuson fedeztek fel hallgatók!



Miért tanuljunk algoritmusokat?

- Felfedezhetők az élet és az univerzum titkai.
- A matematikai modelleket felváltják a számítógépes modellek a tudományos kutatások eredményeként.

$$E = mc^2$$
$$F = ma \quad F = \frac{Gm_1m_2}{r^2}$$
$$\left[-\frac{\hbar^2}{2m} \nabla^2 + V(r) \right] \Psi(r) = E \Psi(r)$$

20th century science
(formula based)

```
for (double t = 0.0; true; t = t + dt)
  for (int i = 0; i < N; i++)
  {
    bodies[i].resetForce();
    for (int j = 0; j < N; j++)
      if (i != j)
        bodies[i].addForce(bodies[j]);
  }
```

21st century science
(algorithm based)

Miért tanuljunk algoritmusokat?

- Szórakoztató és profitot hozhat.

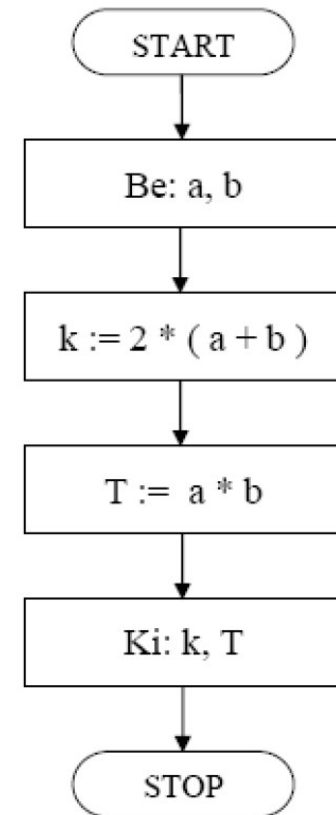
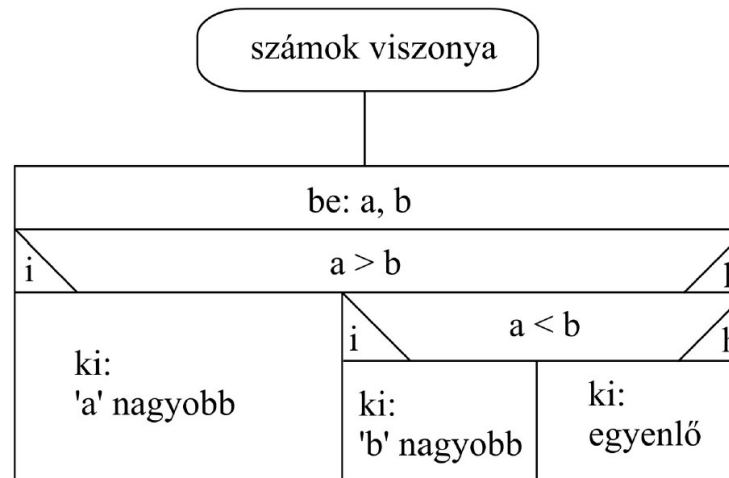


Miért tanulna valami mást?



Algoritmusok ábrázolása, leírása

- Pszeudo kód
 - Mondatszerű leírás
- Folyamatábra
 - Blokkdiagram
- Struktogram
 - Egyetlen téglalap tagolása, amely a teljes feladat részekre bontását jelenti.
- ...



Hogyan tovább? Mi várhat ránk?

Kérdések, felvetések

- Az MI és a robotika nem fog egész iparágakat törölni
- Egészségügy:
 - Orvos: információ feldolgozásra koncentrálni → elemzi az orvosi adatokat, ez alapján diagnózist állít fel → **MI házi orvos az okostelefonon (ALGORITMUSOK)**
 - Ápoló: jó motoros és érzelmi képességekkel rendelkezik → bead egy injekciót, kicserél egy kötést, megfékez egy erőszakos páciens → **nem helyettesítheti egy ápolórobot**
- Az emberek egyre tovább élnek, kevesebb gyerek születik → valószínű **az idős gondozás az egyik leggyorsabban növekvő szektor az emberi munkaerőpiacon (MI-s segítő alkalmazások lehetnek (ALGORITMUSOK))**

Beethoven a gépben

- Egyetlenegy munkahely sem lehet biztonságban az automatizálástól.
- A művészetről általában az emberi érzésekre asszociálunk
 - azt gondoljuk, hogy belső pszichológiai erőket csatornáznak be a művészek
 - a művészet célja, hogy összekössön bennünket az érzelmeinkkel, vagy újakat keltsen bennünk
- **Mi történik akkor, ha az algoritmusok képesek lesznek jobban megérteni és manipulálni az emberi érzelmeket, mint Shakespeare vagy Beyoncé?**
- Az érzelmek biokémiai folyamatok! → **Egy tanuló algoritmus képes lehet elemezni a szenzorai által a testünkből és testünkről közvetített biometrikus adatokat (ALGORITMUSOK) → meghatározzák személyiség típusunkat és hangulatunkat → kiszámítja milyen hatással lesz ránk egy-egy dal**

Beethoven a gépben

- A zene alkalmas arra, hogy a big data elemezze, mivel az input és az output precízen felírható matematikailag:
 - Input = hanghullámok matematikai mintázatai
 - Output = a neurális kisülések elektrokémiai mintázatai
- Csak néhány évtized szükséges ahhoz, hogy több millió zenei élményt végigtanulmányozó algoritmus képes legyen megjósolni, hogy adott input milyen outputot fog kiváltani (ALGORITMUSOK)
- Nincs az DJ, aki vetekedhetne egy ilyen képességgel!?

Beethoven a gépben

- **Mi lesz így az új zenei ízlések és stílusok felfedezésével?**
- No problem! Beállítjuk pl., hogy 8%-a az algoritmus választásainak legyen véletlenszerű, lépjen meg bennünket időnként egy kis funky zenével vagy Puccini operával. (ALGORITMUSOK)
- A reakcióinkat figyelve az MI maga is be tudja állítani a véletlenszerűség mértékét → meg legyen a felfedezés öröme, de ne legyen idegesítő
- **Ha összeveszünk a párunkkal az algoritmus a szomorúságunkat vegye át vagy próbáljon felvidítani?**
 - Vásárlóra hagyjuk a döntést → az algoritmus követi az utasításainkat
 - Utasítjuk az algoritmust, hogy R. Atkinson híres pszichológus tanácsait kövesse
 - Az algoritmus apró igazításokkal a pillanatnyi hangulatunkhoz igazítja a zenét, pl. egy idegesítő résznél átírja a dallamot

Beethoven a gépben

- Gyakran mondják, hogy az embert az fogja meg a művészetekben, ha saját magára talál.
- Meglepő és rémisztő is lehet, az algoritmus lehet, hogy csak nekünk tetsző dallamot ír!
- Általában az tetszik, ami sok embernek tetszik. Lehet táncolni, együtt énekelni stb.
- Millióktól begyűjtött adatokból létrehozott biometrikus adatbázisokat használva az algoritmus mindenkit táncra perdíthet. Egy zenész se lesz erre képes! (ALGORITMUSOK)
- **Mely zeneszerző lesz a jobb, a gépi vagy az emberi?**

Munkák, amelyek eddig nem voltak

- Általános orvosok (legismertebb betegségek diagnosztizálása) helyett → MI orvosok → több pénz juthat pl. sebészeti eljárások kifejlesztésére
- Segíthetjük kiszolgálni, finomítani az MI-t (ALGORITMUSOK)
- Emberi pilóták drónokkal való helyettesítése → új munkahelyek karbantartás, távirányítás, adatelemzés, kiberbiztonság területén
- USA 30 ember szükséges pl. a Szíria felett pilóta nélküli Predatorok és Reaperok működtetéséhez, min. 110 ember kell az adatok elemzéshez → ember hiány az ember nélküli repülőgépekhez?!

Munkák, amelyek eddig nem voltak

- A 2050-es munkaerőpiacot az **ember-MI együttműködés** kell, hogy jellemezze, nem a versengés
- Hatékonyabbak lehetnek a vegyes ember-MI csapatok pl. rendfenntartás, banki szféra
- Deep Blue 1997 legyőzte Garri Kaszparovot → emberek tovább sakkoztak → MI-edzőknek köszönhetően a sakk mesterek jobbak lettek, DE!
- A „kentaúr” becenévre hallgató ember-MI csapatok túltettek az emberen és a számítógépen külön-külön → kinevelhetők a legjobb nyomozók, bankárok, katonák stb.

Problémák lehetnek

- Az új munkahelyek nagy szakértelmet igényelnek → a nagyszámú képzetlen munkaerő nem vagy nehezen talál munkát
- **Átképzések, továbbképzések szükségesek folyamatosan!**
- Pl. ha 2050-ben egy bolti pénztáros vagy gyári munkás elveszíti az állását a robotizálás miatt, nem lesz képes rákkutatóként, drónkezelőként vagy egy ember-MI bankárcsapat tagjaként dolgozni → magas munkanélküliség, képzett munkaerő hiánya → **már most sem ülhetünk a babérjainkon!**

Versengés a köbön

- **Stockfish 8** 2016 számítógépes sakk-világbajnoka: több száz évnyi emberi sakkozói és több évtizednyi számítógépes tapasztalathoz fért hozzá, másodpercenként 70 millió állás átgondolása
- **AlphaZero** (Google) 2017 számítógépes sakk-világbajnoka: csak 80 ezer számítás elvégzésére volt képes, nem tápláltak bele semmiféle sakkstratégiát, **gépi tanulás**i alapelveket alkalmazva tanult meg sakkozni, úgy hogy maga ellen játszott (**ALGORITMUSOK**)
- Stockfish elleni 100 játszmából 28 győzelem és 72 döntetlen
- AlphaZero sok nyerő lépése és stratégiája teljesen szokatlan volt az ember számára
- **AlphaZero 0-ról a mesteri kreativitásig 4 óra jutott el!**

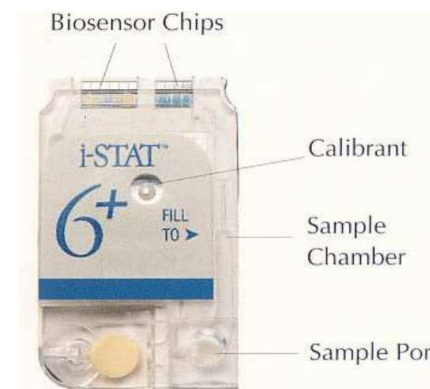
Nem lesz „nyugdíjas szakma”

- Folyamatos átképzés → **érzelmi állóképesség?! → már most stresszjárvány, mi lesz később?!**
- Info- és biotechnológia (külön-külön és összekapcsolódva) kihívásai az elkövetkezendő évtizedekben
- **Jön a „Hallgass az algoritmusokra!” korszak?!**
- Két óriási forradalom összefolyását éljük meg: biológusok fejtik meg az emberi test, az agy és az érzések rejtelseit + informatikusok biztosítanak számunkra korábban nem látott adatfeldolgozási kapacitást → érzéseinket megértő algoritmusok (ALGORITMUSOK)

Info- és biotechnológia

- Néhány évtizeden belül 24 órában monitorozhatják egészségügyi állapotunkat olyan algoritmusok, amelyek folyamatosan testünk biometrikus adataihoz férnek hozzá → pl. influenza, Alzheimer-kór felfedezése az észlelt tünetek előtt → személyiségünkre, DNS-ünkre, fizikumunkra szabott kezelések, életmód és étrend javaslatok → **folyton „betegek” leszünk?!** → folyamatosan javaslatok **(ALGORITMUSOK)**
- **Az egészségügyi algoritmusok jelzéseit figyelembe vesszük vagy sem?**

Info- és biotechnológia

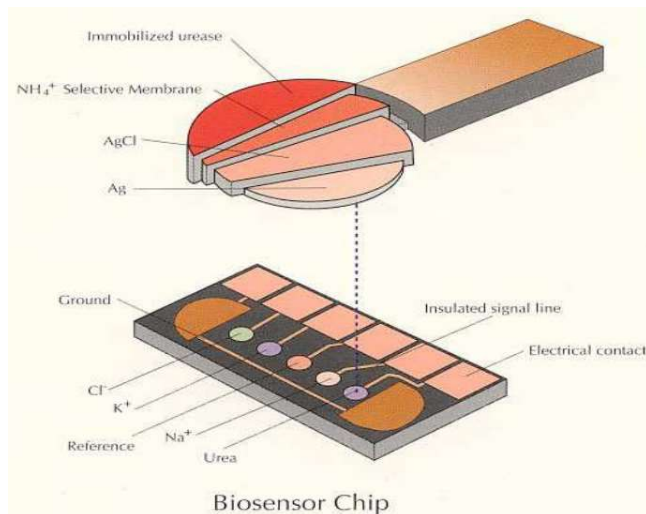


- „Video on demand” – ajánlórendszerek sokasága
- Megadjuk milyen filmeket szeretünk, óriási statisztikai adatbázis segítségével az algoritmus megtalálja a tökéletes választást → **önbevallás megbízhatatlan!**
- **Valós időben adatok gyűjtése**, amikor filmet nézünk (pl. tudni fogja, hogy jónak hirdettük, de fél óránál tovább soha nem tudtuk megnézni)
- A szem és arcizmok mozgása alapján az ember érzelmeinek felismerése (kamera + **algoritmus** → nevetés, sírás stb.)
- **Algoritmus összekapcsolása bioszenzorokkal** → információ a pulzusunkról, vérnyomásunkról, agytevékenységünkről → különbség az erőltetett nevetés és a szívből jövő nevetés között

(ALGORITMUSOK)

Bioszenzorok

- TV (televízió = messze látni) **hamarosan látni fog bennünket?**
→ lehet, hogy ha végig nézünk egy filmet, nem is emlékszünk rá, de pl. a Netflix (aki működteti a tévéalgoritmust) ismerni fogja a személyiségtípusunkat → tudni fogja hogyan hasson ránk → tökéletes film választás → de tovább is léphetünk, akár megmondható mit tanuljunk, hol dolgozzunk stb. (ALGORITMUSOK)



Collect just 2-3 drops of fresh whole blood and fill cartridge.



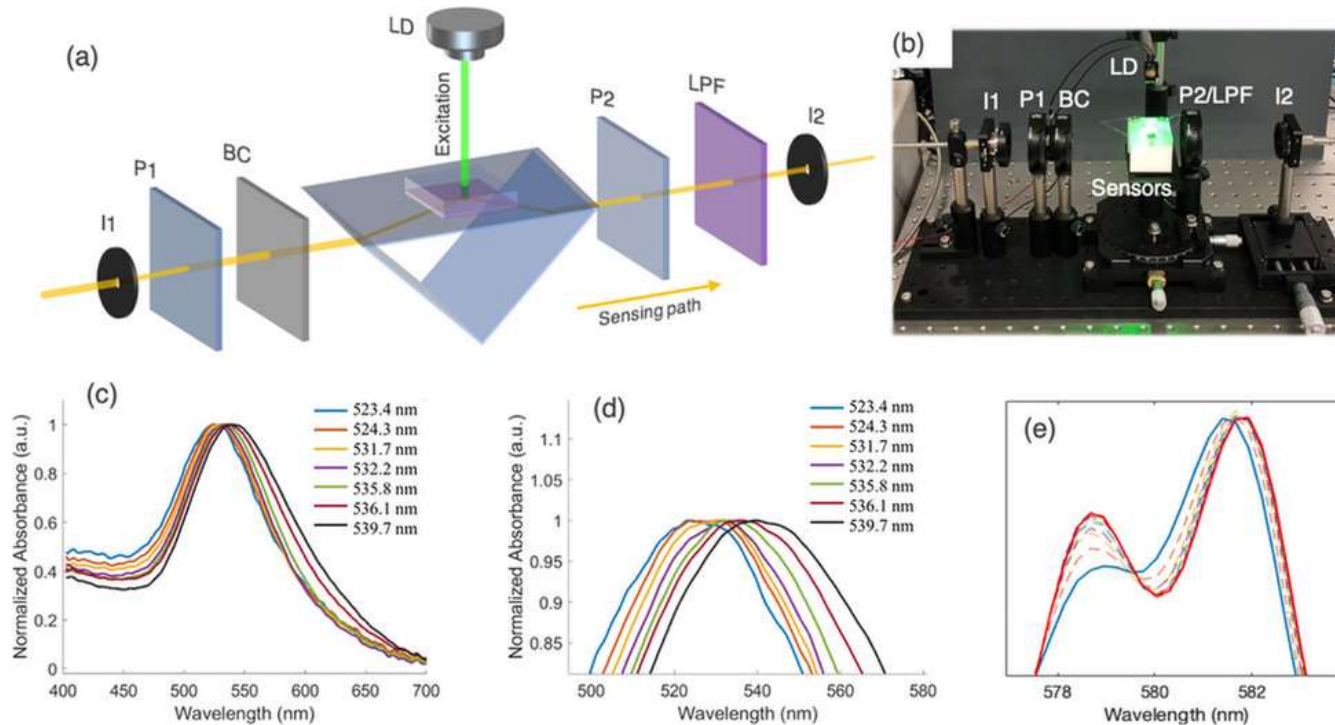
Insert cartridge into the portable, battery-powered analyzer.



View quantitative test results in less than 2 minutes.

Bioszenzorok

- **Optikai bioszenzor detektálja és méri a koronavírus (Svájc):** az új eszköz optikai és egyben hőérzékelő szenzor is, a lelke pedig egy különleges, arany nanostruktúrákat tartalmazó bevonat, amit a szenzor üvegére visznek föl. (ALGORITMUSOK)



Filozofáló autók

- Ha az önvezető autót beprogramozzuk, hogy álljon meg segíteni a bajba jutott idegeneknek, akkor bármi is történik, meg fog állni.
- Ha beprogramozzuk, hogy térjen át a másik sávba, ha gyerekek tévednek eléje, át fog menni, még akkor is, ha ott egy teherautó jön szembe.
- **Mit tegyen? Mentsen meg minket vagy a gyerekeket?** → filozófiai algoritmusok – nem lesznek tökéletesek, de lehetnek jobbak, mint mi → filozófusokra is szükség lesz (ALGORITMUSOK)

Út a magasabb intelligenciához

- A repülőgépek gyorsabban repülnek a madaraknál tollak nélkül
- A számítógépek jobban képesek megoldani problémákat az emlősöknél érzések nélkül
- Az MI-nek tudnia kell **pontosan elemezni az emberi érzéseket** ahhoz, hogy
 - betegségeket kezelhessen,
 - terroristákat azonosíthasson,
 - társakat javasolhasson,
 - irányíthassa egy utca forgalmát,
 - stb.
- Viszont az algoritmusnak nem kell örömet vagy haragot éreznie, hogy felismerje a hazatérő gazdáját meglátó örvendező vagy haragos kutyas eltérő kémiai vagy gesztusmintázatait **(ALGORITMUSOK)**

Mi történik a sok adattal?

Csak egy példa a változásra:

- A reklámpiar csődje jön, ha az algoritmusok választanak és vásárolnak helyettünk?
- Mi történik akkor, ha a Google-től ezt kérdezzük:
„Szia, figyelembe véve mindazt, amit az autókról tudsz és mindazt, amit rólam – beleértve a szokásaimat, a nézeteimet, az igényeimet, a nézeteimet a klímaváltozásról vagy a bevándorlásról -, melyik autó lenne nekem a legjobb?”
- Ha jó választ ad nekünk, mi pedig tapasztalatból megtanultuk, hogy hallgassunk rá, akkor a reklámoknak semmi haszna sem lesz! (ALGORITMUSOK)

Konklúzió

- Ha ma nem is tudjuk, hogyan váltsuk az adatokat készpénzre, akkor is érdemes megtartani azokat, mert az élet irányításának és alakításának a kulcsát rejthetik a jövőben.
- A hétköznapi ember ezt támogatja is minden oldalról, ahol csak lehet adatot közöl, ad át pl. ingyenes szolgáltatásért vagy kutyusos videóért cserébe!?
- Legfontosabb kérdés lehet: **Hogyan szabályozzuk az adatok tulajdonjogát?**

További kérdések

- Mi a teendő, ha olyan helyzettel kerülünk szembe, amelyenre még sohasem volt példa?
- Mit tegyünk, ha eláraszt bennünket a hatalmas mennyiségű információ és reménytelen, hogy mindet befogadjuk és elemezzük?
- Hogyan kell élni egy olyan világban, amelynek a bizonytalanság nem működési hibája, hanem meghatározó vonása?
- **Komoly mentális rugalmasságra és erős érzelmi egyensúlyra lesz szükség.**
- **Képesnek kell lenni újra és újra elengedni azt, amit ismerünk, és otthon érezni magunkat az ismeretlenben.**
- **Fontos megismerni ÖNMAGAD!**

Az algoritmusok szemmel tartanak!

- Kevés tudással rendelkezünk, még ha úgy is gondoljuk, hogy mindent tudunk! Ez a kevés tudás is elavul! → folyamatos fejlődés, kutatás, nem tárgyi tudás megszerzése, hanem elemző, probléma megoldó képesség fejlesztése
- A technológia nem rossz, ha tudod, mit akarsz az életben, a technológia segíthet elérni azt.
- Ha nem tudod mit akarsz az életben, megeshet, hogy a technológia alakítja a céljaidat és átveszi az uralmat az életed felett.
- Láttál már az utcán okostelefonra szegezett tekintettel őgyelgő zombikat? Szerinted ők irányítják a technológiát vagy az őket?
- Magadat kell megismerni – mire vagy képes, mit akarsz, legyenek célok -, hogy tényleg a „szívedre hallgass” és ne a Coca-Cola reklámra vagy hasonlókra.

Néhány egyszerű, de hatékony megoldás - algoritmus

Ackermann függvény

avagy nem minden olyan egyszerű, mint amilyennek látszik

Definíció: Legyen $A(x, y)$ a következő rekurzív módon definiált függvény:

$$\begin{aligned} A(0, y) &:= y + 1 && y \geq 0 \\ A(x, 0) &:= A(x - 1, 1) && x \geq 1 \\ A(x, y) &:= A(x - 1, A(x, y - 1)) && x, y \geq 1 \end{aligned}$$

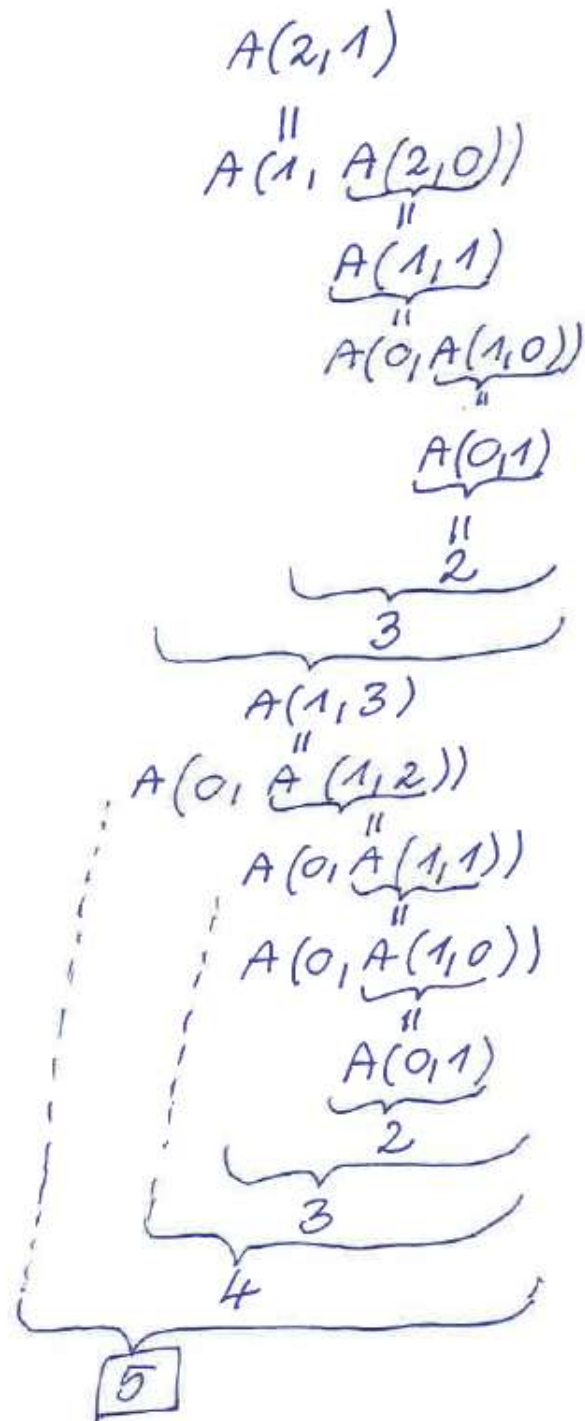
Határozzuk meg pl. $A(4, 2)$ és $A(5, 1)$ értékét!

Ackermann függvény C#

```
using System;
namespace Ackermann_fuggveny
{
    class Ackermann
    {
        public static int A(int x, int y)
        {
            if (x==0)
                return y+1;
            else if (y==0)
                return A(x-1,1);
            else
                return A(x-1, A(x,y-1));
        }
        static void Main(string[] args)
        {
            System.Console.WriteLine(A(3, 8));
        }
    }
}
```

$$\begin{aligned}
 A(0, y) &:= y + 1 \quad y \geq 0 \\
 A(x, 0) &:= A(x-1, 1) \quad x \geq 1 \\
 A(x, y) &:= A(x-1, A(x, y-1)) \\
 &\quad x, y \geq 1
 \end{aligned}$$

$$\begin{aligned}
 A(3, 1) &\Rightarrow 13 \\
 A(3, 2) &\Rightarrow 29 \\
 A(3, 3) &\Rightarrow 61 \\
 A(3, 8) &\Rightarrow 2045 \\
 A(4, 1) &\Rightarrow 65533 \\
 A(4, 2) &\Rightarrow \text{már nem megy!}
 \end{aligned}$$



Egy másik hasznos, de egyszerű megoldás

Postfix konverzió

Infix jelölés: $(a + b) * (c - d) + a$ zárójelezés szükséges

Lukasiewicz: nincs szükség zárójelezésre \Rightarrow **postfix jelölés**

Jelölje $\Phi(K)$ a K kifejezés postfix alakját.

$$\Phi((a + b) * (c - d) + a) = ab + cd - *a +$$

Probléma: Postfix konverzió

Bemenet: K aritmetikai kifejezés infix jelölésben

Kimenet: K postfix alakja

A szabályos aritmetikai kifejezések megadhatók **rekurzív szintaxis diagramokkal**.

Postfix konverzió

Minden kifejezés vagy egy *tag*, vagy *tagok additív műveleti jelekkel elválasztott sorozata*.

Minden kifejezés egyértelműen felbontható

$K = t_1 \oplus_1 t_2 \dots \oplus_m t_{m+1}$ alakban, ahol $\oplus_i \in \{+, -\}$ és t_i tag.

Ekkor $\Phi(K) = \Phi(t_1)\Phi(t_2) \oplus_1 \dots \Phi(t_{m+1})\oplus_m$

Postfix konverzió

Minden **tag** vagy egy **tényező**, vagy **tényezők multiplikatív műveleti jellel elválasztott sorozata**.

Minden tag egyértelműen felbontható $T = t_1 \otimes_1 t_2 \dots \otimes_m t_{m+1}$ alakban, ahol $\otimes_i \in \{*, /\}$ és t_i tényező.

Ekkor $\Phi(T) = \Phi(t_1)\Phi(t_2) \otimes_1 \dots \Phi(t_{m+1})\otimes_m$

Minden **tényező** vagy **zárójelbe tett kifejezés**; (K) és $\Phi((K)) = \Phi(K)$, vagy **azonosító**; a és $\Phi(a) = a$

```

using System;
using System.Collections.Generic;
using System.Text;

namespace Postfix
{
    public class Postfix
    {
        private static bool Jo = true;
        private static String S;
        private static int i = 0;
        private static char Jel;
        private static String Postform;

        private static void KovJel()
        {
            Jel = S[i];
            i++;
        }
    }
}

```

```

public static void Kifejezes()
{
    char M;
    Tag();
    while (Jo && (Jel == '+' || Jel == '-'))
    {
        M = Jel;
        KovJel();
        Tag();
        Postform = Postform + M;
    }
}

private static void Tag()
{
    char M;
    Tenyezo();
    while (Jo && (Jel == '*' || Jel == '/'))
    {
        M = Jel;
        KovJel();
        Tenyezo();
        Postform = Postform + M;
    }
}
}

```

```

private static void Tenyezo()
{
    if ('a' <= Jel && Jel <= 'z')
    {
        Postform = Postform + Jel;
        KovJel();
    }
    else if (Jel == '(')
    {
        KovJel();
        Kifejezes();
        if (Jo && Jel == ')')
            KovJel();
        else
            Jo = false;
    }
    else
        Jo = false;
}

```

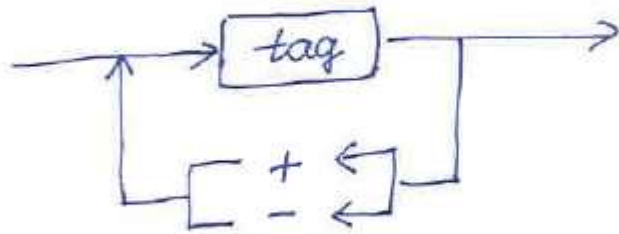
```

public static String postfix(String K)
{
    S = K + '.';
    KovJel();
    Postform = "";
    Kifejezes();
    return Postform;
}

static void Main(string[] args)
{
    System.Console.WriteLine("Az átalakítandó
műveletsorozat: a+b*(a-b)/(x+y)");
    System.Console.Write("Átalakítás után a postfix
alak: ");
    System.Console.WriteLine(postfix("(a+b)*(a-b)"));
}
}

```

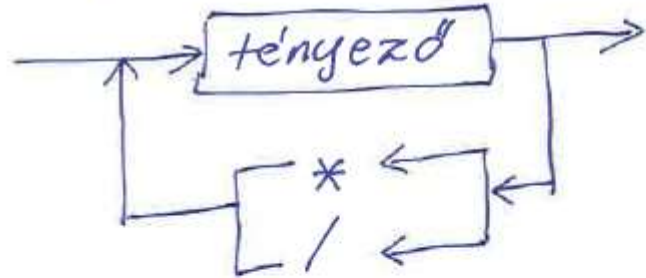
Kifejezés:



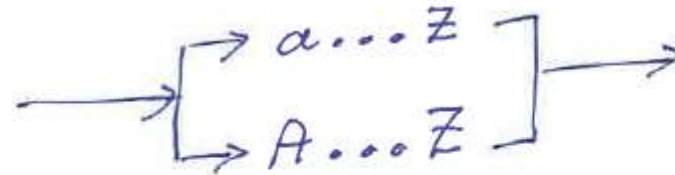
Tényező:



Tag:



Azonosító:



Példák: $(a+b) / (c-d) = ab + cd - /$

$((a+b) * c + d) / (e + f + g) = ab + c * d + e f + g + /$