

# Paraméteres bonyolultság

Kovács Milán, Nemkin Viktória

2021. március 16.

1 Motiváció

2 Bar Fight Prevention problem

# P nyelvosztály definíciója

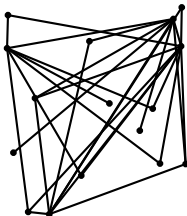
A  $P$  azoknak a nyelveknek az osztálya, amelyekhez van polinom időkorlátos algoritmus (determinisztikus Turing-gép), azaz ha létezik olyan  $p(n)$  polinom, hogy az algoritmus **az  $n$  méretű bemeneteken legfeljebb  $p(n)$  lépést tesz.**

Szeretnénk minden problémára polinom időkorlátos algoritmusokat adni...

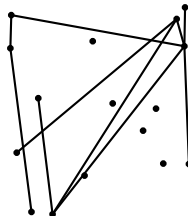
Kérdés: Miért csak a bemenet hosszára figyelünk?

## Példa: Sűrű / ritka gráfok

Sűrű gráf (TODO: Egy kevésbé csúnya gráf.)



Ritka gráf (TODO: Egy kevésbé csúnya gráf.)



Erre a két gráfra nézzünk gráfalgoritmusokat:

- Legnagyobb független csúcshalmaz.
- Csúcsszínezés.
- Stb...

Mindkét gráfban ugyanannyi csúcs van, ezért ha szomszédossági mátrixukkal adjuk meg őket, akkor ugyanakkora lesz az input mérete, azonban a 2. gráfban a fenti kérdésekre elég hamar választ tudunk adni.

## Példa: Prímtényezős felbontás

Feladat: számok prímtényezős felbontását megadni.

$$4503599627370496 = 2^{52}$$

$$1125897758834689 = 524287 \cdot 2147483647$$

Ugyanolyan sok számjegyből állnak a számok, tehát ugyanolyan hosszú az input méretünk, mégis az elsőt nagyon gyorsan meg lehet találni, a másodikat sokkal lassabban.

# Valós példák

Nagyon sok NP-beli probléma előjön a való életben és nagyon jó lenne őket megoldani.  
A gyakorlatban sokszor nem általános megoldásokat kell adni, általában vannak korlátok...

- Facebook gráf: Fokszám kicsi.
- ...?

# Feladat

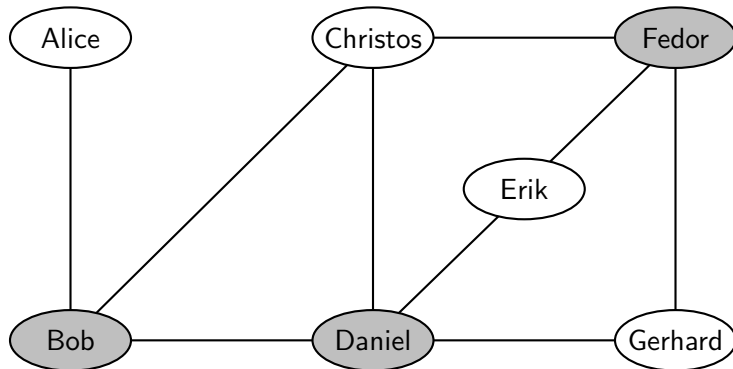
Képzeljük el, hogy biztonsági őrként dolgozunk egy falusi bárban. Péntekenként nagy tömeg szokott lenni és általában bunyóban végződik a történet... A mi feladatunk kidobni az ittas vendégeket, ami nagyon fárasztó és nem túl mókás. Elhatározzuk, hogy megelőző intézkedéseket teszünk...

Mindenkit ismerünk a faluban és azt is tudjuk ki kivel nincs jóban, kik fognak várhatóan összeverekedni. A tervünk tehát az, hogy csak olyan embereket engedünk be a bárba, akik jóban vannak egymással, így elkerüljük a verekedést.

Azonban a bár menedzsmentje maximalizálni akarja a profitot, ezért azt a kikötést teszi, hogy legfeljebb  $k$  darab vendéget lehet elutasítani az ajtóban.

A feladat tehát a következő: Ismerjük a bárba bejövő emberek listáját ( $n$  ember), minden emberpárra tudjuk, hogy fognak-e verekedni ha mindkettőjüket beengedjük. Ki kell találni, hogy be lehet-e úgy engedni az embereket, hogy legfeljebb  $k$  darab embert utasítunk el, úgy bent ne törjön ki verekedés.

## Példa: a szürkétet kell kidobni





# Brute force megoldás

- Brute force algoritmus.
- Minden lehetséges részhalmazt megnézzük: ha őket dobniuk ki a többiek verekednének-e?
- $2^n$ , pl  $n=1000$ -re már túl nagy.

# Ha tudjuk, hogy a $k$ kicsi, pl. $k \leq 10$

- A menedzsment úgysem fog nagy  $k$ -t engedni.
- Aki 0 fokszámú azt beengedhetem, mert senkivel nem fog összeveszni.
- Aki  $k$ -nál nagyobb fokszámú azt nem engedhetem be, mert akkor a szomszédjait kellene kitiltani, akik  $k$ -nál többen vannak.
- Ha valakit kitiltok akkor  $k$ -t csökkentem eggyel.
- Maradék gráf:  $1 \dots k$  fokú csúcsok. Minden kitiltás így  $k$  vagy kevesebb konfliktust fog megoldani a továbbiakban.
- Ha több mint  $k^2$  élünk van akkor biztosan nem megoldható a feladat, készen vagyunk.
- Ha  $k^2$  vagy kevesebb élünk van, akkor legfeljebb  $2k^2$  csúcsunk lehet (minden élnek két vége van és nincs 0 fokú csúcs).
- $\binom{2k^2}{k}$  mostmár  $k \leq 10$ -re már jobb mint az előbbi  $2^n$ .

# 1 fokú csúcsok

- Az 1 fokú csúcs és szomszédja esetében: ha beengedem a csúcsot akkor az 1 darab szomszédját nem engedhetem be.
- Ezzel biztos nem lett rosszabb a helyzet, mert ha a csúcsot nem engedem be akkor a szomszédját beengedhetem, de annak még lehetnek egyéb szomszédjai is.
- Ezért engedjük be az 1 fokú csúcsokat és tiltsuk ki a szomszédokat (ezzel  $k$ -t is csökkentjük 1-el).
- Így mostmár  $2..k$  konfliktus lehet.
- Erre megint kiszámolom a max csúcsszámot, ez mostmár csak  $k^2$ , erre még jobb szám jön ki.



Itt van még a példának folytatása bounded search tree-kkel, de azt inkább Milánnak kellene elmondania.

# Kernelizációs technika általánosan

# Vertex cover feladat megoldása egyben

# Paraméteres komplexitás definíciója általánosan

# Szemezgetés