# Algoritmuselmélet

## Függvények nagyságrendje, elágazás és korlátozás, dinamikus programozás

Katona Gyula Y.

Számítástudományi és Információelméleti Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

1. előadás

# Algoritmus fogalma

- Egyelőre nem definiáljuk rendesen az algoritmus fogalmát.

# Algoritmus fogalma

- Egyelőre nem definiáljuk rendesen az algoritmus fogalmát.
- Eljárás, recept, módszer.

# Algoritmus fogalma

- Egyelőre nem definiáljuk rendesen az algoritmus fogalmát.
- Eljárás, recept, módszer.
- Jól meghatározott lépések egymásutánja, amelyek már elég pontosan, egyértelműen megfogalmazottak ahhoz, hogy gépiesen végrehajthatók legyenek.

# A szó eredete

Al Khvarizmi (Mohamed ibn Músza) bagdadi matematikus a IX. században könyvet írt az egészekkel való alapműveletek végzéséről.

# A szó eredete

Al Khvarizmi (Mohamed ibn Músza) bagdadi matematikus a IX. században könyvet írt az egészekkel való alapműveletek végzéséről.

algoritmus $\leftrightarrow$ számítógép program

# A szó eredete

Al Khvarizmi (Mohamed ibn Músza) bagdadi matematikus a IX. században könyvet írt az egészekkel való alapműveletek végzéséről.

algoritmus $\leftrightarrow$ számítógép program

valós probléma

# A szó eredete

Al Khvarizmi (Mohamed ibn Músza) bagdadi matematikus a IX. században könyvet írt az egészekkel való alapműveletek végzéséről.

$$\text{algoritmus} \leftrightarrow \text{számítógép program}$$

valós probléma $\Longrightarrow$ absztrakt modell

# A szó eredete

Al Khvarizmi (Mohamed ibn Músza) bagdadi matematikus a IX. században könyvet írt az egészekkel való alapműveletek végzéséről.

algoritmus $\leftrightarrow$ számítógép program

valós probléma $\Longrightarrow$ absztrakt modell $\Longrightarrow$ algoritmus

# A szó eredete

Al Khvarizmi (Mohamed ibn Músza) bagdadi matematikus a IX. században könyvet írt az egészekkel való alapműveletek végzéséről.

$$\text{algoritmus} \leftrightarrow \text{számítógép program}$$

valós probléma $\Longrightarrow$ absztrakt modell $\Longrightarrow$ algoritmus $\Longrightarrow$ program

Cél: feladatokra hatékony eljárás kidolgozása

# A szó eredete

Al Khvarizmi (Mohamed ibn Músza) bagdadi matematikus a IX. században könyvet írt az egészekkel való alapműveletek végzéséről.

algoritmus $\leftrightarrow$ számítógép program

valós probléma $\Longrightarrow$ absztrakt modell $\Longrightarrow$ algoritmus $\Longrightarrow$ program

Cél: feladatokra hatékony eljárás kidolgozása

Hatékony $\Longrightarrow$ gyors, kevés memória, kevés tárhely

# Milyen hatékony egy algoritmus?

- Legtöbbször csak a lépésszám nagyságrendje érdekes. Hogyan függ a lépésszám az input méretétől?

# Milyen hatékony egy algoritmus?

- Legtöbbször csak a lépésszám nagyságrendje érdekes.
  Hogyan függ a lépésszám az input méretétől?
- Az input méretét legtöbbször $n$-nel jelöljük.

# Milyen hatékony egy algoritmus?

- Legtöbbször csak a lépésszám nagyságrendje érdekes. Hogyan függ a lépésszám az input méretétől?
- Az input méretét legtöbbször $n$-nel jelöljük.
- A lépésszám ennek egy $f$ függvénye, azaz ha $n$ méretű az input, akkor az algoritmus $f(n)$ lépést végez.

# Milyen hatékony egy algoritmus?

- Legtöbbször csak a lépésszám nagyságrendje érdekes. Hogyan függ a lépésszám az input méretétől?
- Az input méretét legtöbbször $n$-nel jelöljük.
- A lépésszám ennek egy $f$ függvénye, azaz ha $n$ méretű az input, akkor az algoritmus $f(n)$ lépést végez.
- Igazából az $f$ függvény az érdekes.

# Milyen hatékony egy algoritmus?

- Legtöbbször csak a lépésszám nagyságrendje érdekes. Hogyan függ a lépésszám az input méretétől?
- Az input méretét legtöbbször $n$-nel jelöljük.
- A lépésszám ennek egy $f$ függvénye, azaz ha $n$ méretű az input, akkor az algoritmus $f(n)$ lépést végez.
- Igazából az $f$ függvény az érdekes.
- $100n$ vagy $101n$, általában mindegy

# Milyen hatékony egy algoritmus?

- Legtöbbször csak a lépésszám nagyságrendje érdekes.
  Hogyan függ a lépésszám az input méretétől?
- Az input méretét legtöbbször $n$-nel jelöljük.
- A lépésszám ennek egy $f$ függvénye, azaz ha $n$ méretű az input, akkor az algoritmus $f(n)$ lépést végez.
- Igazából az $f$ függvény az érdekes.
- $100n$ vagy $101n$, általában mindegy
- $n^2$ vagy $n^3$ már sokszor nagy különbség, de néha mindegy

# Milyen hatékony egy algoritmus?

- Legtöbbször csak a lépésszám nagyságrendje érdekes. Hogyan függ a lépésszám az input méretétől?
- Az input méretét legtöbbször $n$-nel jelöljük.
- A lépésszám ennek egy $f$ függvénye, azaz ha $n$ méretű az input, akkor az algoritmus $f(n)$ lépést végez.
- Igazából az $f$ függvény az érdekes.
- $100n$ vagy $101n$, általában mindegy
- $n^2$ vagy $n^3$ már sokszor nagy különbség, de néha mindegy
- $n^2$ vagy $2^n$ már mindig nagy különbség

# Például

## Kérdés

*Egy $10^{10}$ művelet/mp sebességű számítógép mennyi ideig dolgozik, ha $f(n)$ műveletet kell végrehajtani $n$ méretű bemenetre?*

| $n$ | $f(n)$ $n$ | $n^2$ | $\log_{10} n$ | $2^n$ | $n!$ |
|---|---|---|---|---|---|
| 10 | $10^{-9}$ | $10^{-8}$ | $10^{-10}$ | $1{,}02 \cdot 10^{-7}$ | $3{,}6 \cdot 10^{-4}$ |
| $10^2$ | $10^{-8}$ | $10^{-6}$ | $2 \cdot 10^{-10}$ | $4 \cdot 10^{12}$ év | $2.9{,}\cdot 10^{140}$ év |
| $10^6$ | $10^{-4}$ | 100 | $6 \cdot 10^{-10}$ | $3{,}1 \cdot 10^{301.012}$ év | $2{,}6 \cdot 10^{5.565.691}$ év |
| $10^9$ | 0,1 | 3,1 év | $9 \cdot 10^{-9}$ | sok év | sok év |

# Függvények nagyságrendje

**Definíció**

*Ha $f(n)$ és $g(n)$ az $\mathbb{R}^+$ egy részhalmazán értelmezett, valós értékeket felvevő függvények, akkor $f = O(g)$ jelöli azt a tényt, hogy vannak olyan $c, n_0 > 0$ állandók, hogy $|f(n)| \leq c|g(n)|$ teljesül, ha $n \geq n_0$.*

# Példák

- $100n + 300 = O(n)$
  Biz: $100n + 300 \leq 100n + n \leq 101n \leq cn$, ha $n \geq 300, \; c = 101$

# Példák

- $100n + 300 = O(n)$
  Biz: $100n + 300 \leq 100n + n \leq 101n \leq cn$, ha $n \geq 300,\ c = 101$
- $5n^2 + 3n = O(n^2)$
  Biz: $5n^2 + 3n \leq 5n^2 + 3n^2 \leq 8n^2 \leq cn^2$, ha $n \geq 100,\ c = 8$

# Példák

- $100n + 300 = O(n)$
  Biz: $100n + 300 \leq 100n + n \leq 101n \leq cn$, ha $n \geq 300,\ c = 101$
- $5n^2 + 3n = O(n^2)$
  Biz: $5n^2 + 3n \leq 5n^2 + 3n^2 \leq 8n^2 \leq cn^2$, ha $n \geq 100,\ c = 8$
- $n^4 + 5n^3 = O(n^5)$
  Biz: $n^4 + 5n^3 \leq 6n^4 \leq n^5 \leq cn^5$, ha $n \geq 6,\ c = 1$

# Példák

- $n^{1000} = O(2^n)$
  Biz: Teljes indukcióval, legyen $c = 1$, $n_0 = 10^6$.

# Példák

- $n^{1000} = O(2^n)$
  Biz: Teljes indukcióval, legyen $c = 1$, $n_0 = 10^6$.
  $n = 10^6$-re igaz, mert $10^{6000} \leq (2^4)^{6000} \leq 2^{10^6}$.

# Példák

- $n^{1000} = O(2^n)$
  Biz: Teljes indukcióval, legyen $c = 1$, $n_0 = 10^6$.
  $n = 10^6$-re igaz, mert $10^{6000} \leq (2^4)^{6000} \leq 2^{10^6}$.
  Tegyük fel, hogy $k$-ra igaz.

# Példák

- $n^{1000} = O(2^n)$
  Biz: Teljes indukcióval, legyen $c = 1$, $n_0 = 10^6$.
  $n = 10^6$-re igaz, mert $10^{6000} \leq (2^4)^{6000} \leq 2^{10^6}$.
  Tegyük fel, hogy $k$-ra igaz.
  Felhasználjuk, hogy ha $k \geq 10^6$ akkor
  $\binom{1000}{i} \leq 1000^i = 1000 \cdot 1000^{i-1} \leq 1000^{i-1} \cdot 1000^{i-1} =$
  $= (10^6)^{i-1} \leq k^{i-1}$.

# Példák

- $n^{1000} = O(2^n)$
  Biz: Teljes indukcióval, legyen $c = 1, n_0 = 10^6$.
  $n = 10^6$-re igaz, mert $10^{6000} \le (2^4)^{6000} \le 2^{10^6}$.
  Tegyük fel, hogy $k$-ra igaz.
  Felhasználjuk, hogy ha $k \ge 10^6$ akkor
  $\binom{1000}{i} \le 1000^i = 1000 \cdot 1000^{i-1} \le 1000^{i-1} \cdot 1000^{i-1} =$
  $= (10^6)^{i-1} \le k^{i-1}$.
  $(k+1)^{1000} = k^{1000} + \cdots + \binom{1000}{i} k^{1000-i} + \cdots \le k^{1000} + \cdots +$
  $k^{i-1}k^{1000-i} + \cdots \le k^{1000} + 1000k^{999} \le 2 \cdot k^{1000} \le 2 \cdot 2^k = 2^{k+1}$,
  ha $k \ge 10^6$.

# Példák

- $n^{1000} = O(2^n)$
  Biz: Teljes indukcióval, legyen $c = 1, n_0 = 10^6$.
  $n = 10^6$-re igaz, mert $10^{6000} \leq (2^4)^{6000} \leq 2^{10^6}$.
  Tegyük fel, hogy $k$-ra igaz.
  Felhasználjuk, hogy ha $k \geq 10^6$ akkor
  $\binom{1000}{i} \leq 1000^i = 1000 \cdot 1000^{i-1} \leq 1000^{i-1} \cdot 1000^{i-1} =$
  $= (10^6)^{i-1} \leq k^{i-1}$.
  $(k+1)^{1000} = k^{1000} + \cdots + \binom{1000}{i} k^{1000-i} + \cdots \leq k^{1000} + \cdots +$
  $k^{i-1} k^{1000-i} + \cdots \leq k^{1000} + 1000k^{999} \leq 2 \cdot k^{1000} \leq 2 \cdot 2^k = 2^{k+1}$,
  ha $k \geq 10^6$.
- $\log_2^{1000}(n) = O(n)$

# Példák

- $n^{1000} = O(2^n)$
  Biz: Teljes indukcióval, legyen $c = 1, n_0 = 10^6$.
  $n = 10^6$-re igaz, mert $10^{6000} \leq (2^4)^{6000} \leq 2^{10^6}$.
  Tegyük fel, hogy $k$-ra igaz.
  Felhasználjuk, hogy ha $k \geq 10^6$ akkor
  $\binom{1000}{i} \leq 1000^i = 1000 \cdot 1000^{i-1} \leq 1000^{i-1} \cdot 1000^{i-1} =$
  $= (10^6)^{i-1} \leq k^{i-1}$.
  $(k+1)^{1000} = k^{1000} + \cdots + \binom{1000}{i} k^{1000-i} + \cdots \leq k^{1000} + \cdots +$
  $k^{i-1} k^{1000-i} + \cdots \leq k^{1000} + 1000k^{999} \leq 2 \cdot k^{1000} \leq 2 \cdot 2^k = 2^{k+1}$,
  ha $k \geq 10^6$.
- $\log_2^{1000}(n) = O(n)$
  Biz: Mivel a logaritmus függvény monoton nő, vehetjük a fentiek logaritmusát.

# Példák

- $n^{1000} = O(2^n)$
  Biz: Teljes indukcióval, legyen $c = 1, n_0 = 10^6$.
  $n = 10^6$-re igaz, mert $10^{6000} \leq (2^4)^{6000} \leq 2^{10^6}$.
  Tegyük fel, hogy $k$-ra igaz.
  Felhasználjuk, hogy ha $k \geq 10^6$ akkor
  $\binom{1000}{i} \leq 1000^i = 1000 \cdot 1000^{i-1} \leq 1000^{i-1} \cdot 1000^{i-1} =$
  $= (10^6)^{i-1} \leq k^{i-1}$.
  $(k+1)^{1000} = k^{1000} + \cdots + \binom{1000}{i} k^{1000-i} + \cdots \leq k^{1000} + \cdots +$
  $k^{i-1} k^{1000-i} + \cdots \leq k^{1000} + 1000k^{999} \leq 2 \cdot k^{1000} \leq 2 \cdot 2^k = 2^{k+1}$,
  ha $k \geq 10^6$.
- $\log_2^{1000}(n) = O(n)$
  Biz: Mivel a logaritmus függvény monoton nő, vehetjük a fentiek logaritmusát.
- $2^n = O(n!)$

# Példák

- $n^{1000} = O(2^n)$
  Biz: Teljes indukcióval, legyen $c = 1$, $n_0 = 10^6$.
  $n = 10^6$-re igaz, mert $10^{6000} \leq (2^4)^{6000} \leq 2^{10^6}$.
  Tegyük fel, hogy $k$-ra igaz.
  Felhasználjuk, hogy ha $k \geq 10^6$ akkor
  $\binom{1000}{i} \leq 1000^i = 1000 \cdot 1000^{i-1} \leq 1000^{i-1} \cdot 1000^{i-1} =$
  $= (10^6)^{i-1} \leq k^{i-1}$.
  $(k+1)^{1000} = k^{1000} + \cdots + \binom{1000}{i} k^{1000-i} + \cdots \leq k^{1000} + \cdots +$
  $k^{i-1} k^{1000-i} + \cdots \leq k^{1000} + 1000k^{999} \leq 2 \cdot k^{1000} \leq 2 \cdot 2^k = 2^{k+1}$,
  ha $k \geq 10^6$.
- $\log_2^{1000}(n) = O(n)$
  Biz: Mivel a logaritmus függvény monoton nő, vehetjük a fentiek logaritmusát.
- $2^n = O(n!)$
- $n! = O(n^n)$

# Példák

Igaz-e, hogy $n^2 = O(n)$?

# Példák

Igaz-e, hogy $n^2 = O(n)$?

Nem.

Biz: Indirekt, tegyük fel, hogy létezik olyan $c, n_0$, hogy $n^2 \leq cn$ teljesül minden $n \geq n_0$ esetén.

# Példák

Igaz-e, hogy $n^2 = O(n)$?

Nem.
Biz: Indirekt, tegyük fel, hogy létezik olyan $c, n_0$, hogy $n^2 \leq cn$ teljesül minden $n \geq n_0$ esetén.
Ekkor $n \leq c$ teljesül minden $n \geq n_0$ esetén, ami nyilván nem igaz, ha $n > c$.

# Függvények nagyságrendje

**Definíció**

*Ha $f(n)$ és $g(n)$ az $\mathbb{R}^+$ egy részhalmazán értelmezett, valós értékeket felvevő függvények, akkor $f = \Omega(g)$ jelöli azt a tényt, hogy vannak olyan $c, n_0 > 0$ állandók, hogy $|f(n)| \geq c|g(n)|$ teljesül, ha $n \geq n_0$.*

# Függvények nagyságrendje

## Definíció

*Ha $f(n)$ és $g(n)$ az $\mathbb{R}^+$ egy részhalmazán értelmezett, valós értékeket felvevő függvények, akkor $f = \Omega(g)$ jelöli azt a tényt, hogy vannak olyan $c, n_0 > 0$ állandók, hogy $|f(n)| \geq c|g(n)|$ teljesül, ha $n \geq n_0$.*

Például:

- $100n - 300 = \Omega(n)$, hiszen $n > 300,\ c = 99$-re teljesülnek a feltételek

# Függvények nagyságrendje

**Definíció**

*Ha $f(n)$ és $g(n)$ az $\mathbb{R}^+$ egy részhalmazán értelmezett, valós értékeket felvevő függvények, akkor $f = \Omega(g)$ jelöli azt a tényt, hogy vannak olyan $c, n_0 > 0$ állandók, hogy $|f(n)| \geq c|g(n)|$ teljesül, ha $n \geq n_0$.*

Például:

- $100n - 300 = \Omega(n)$, hiszen $n > 300,\ c = 99$-re teljesülnek a feltételek
- $5n^2 - 3n = \Omega(n^2)$
- $n^4 - 5n^3 = \Omega(n^4)$
- $2^n = \Omega(n^{1000})$

# Függvények nagyságrendje

**Definíció**

*Ha* $f = O(g)$ *és* $f = \Omega(g)$ *is teljesül, akkor* $f = \Theta(g)$.

# Függvények nagyságrendje

**Definíció**

*Ha $f = O(g)$ és $f = \Omega(g)$ is teljesül, akkor $f = \Theta(g)$.*

Például:

- $100n - 300 = \Theta(n)$

# Függvények nagyságrendje

**Definíció**

*Ha $f = O(g)$ és $f = \Omega(g)$ is teljesül, akkor $f = \Theta(g)$.*

Például:

- $100n - 300 = \Theta(n)$
- $5n^2 - 3n = \Theta(n^2)$
- $n^4 - 5n^3 = \Theta(n^4)$
- $1000 \cdot 2^n = \Theta(2^n)$

# Függvények nagyságrendje

## Definíció

*Legyenek $f(n)$ és $g(n)$ a pozitív egészeken értelmezett, valós értékű függvények. Ekkor az $f = o(g)$ jelöléssel rövidítjük azt, hogy*

$$\frac{f(n)}{g(n)} \to 0, \ \ \textit{ha}\ n \to \infty.$$

# Függvények nagyságrendje

## Definíció

*Legyenek $f(n)$ és $g(n)$ a pozitív egészeken értelmezett, valós értékű függvények. Ekkor az $f = o(g)$ jelöléssel rövidítjük azt, hogy*

$$\frac{f(n)}{g(n)} \to 0, \ \textit{ha } n \to \infty.$$

Például:

- $100n + 300 = o(n^2)$, hiszen $\frac{100n+300}{n^2} \to 0$ ha $n \to \infty$

# Függvények nagyságrendje

## Definíció

*Legyenek $f(n)$ és $g(n)$ a pozitív egészeken értelmezett, valós értékű függvények. Ekkor az $f = o(g)$ jelöléssel rövidítjük azt, hogy*

$$\frac{f(n)}{g(n)} \to 0, \ \textit{ha } n \to \infty.$$

Például:

- $100n + 300 = o(n^2)$, hiszen $\frac{100n+300}{n^2} \to 0$ ha $n \to \infty$
- $5n^2 + 3n = o(n^3)$
- $n^4 + 5n^3 = o(n^4 \log_2 n)$
- $n^{1000} = o(2^n)$

# Algoritmikus problémák megoldása

$$\boxed{\text{Algoritmikus probléma}} \xrightarrow{\mathcal{A}} \boxed{\text{modell}} \xrightarrow{\mathcal{B}} \boxed{\text{program}}$$

# Algoritmikus problémák megoldása

$\boxed{\text{Algoritmikus probléma}} \xrightarrow{\mathcal{A}} \boxed{\text{modell}} \xrightarrow{\mathcal{B}} \boxed{\text{program}}$

$\mathcal{A}$: pontosítás, egyszerűsítés, absztrakció, lényegtelen elemek kiszűrése, a lényeg kihámozása

# Algoritmikus problémák megoldása

$$\boxed{\text{Algoritmikus probléma}} \xrightarrow{\mathcal{A}} \boxed{\text{modell}} \xrightarrow{\mathcal{B}} \boxed{\text{program}}$$

$\mathcal{A}$: pontosítás, egyszerűsítés, absztrakció, lényegtelen elemek kiszűrése, a lényeg kihámozása

Modell: sokféle lehet, elég tág, de elég egyszerű, formalizált, pontos

# Algoritmikus problémák megoldása

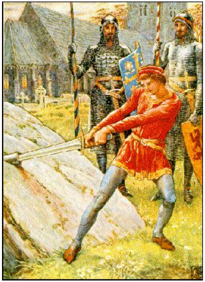
$$\boxed{\text{Algoritmikus probléma}} \xrightarrow{\mathcal{A}} \boxed{\text{modell}} \xrightarrow{\mathcal{B}} \boxed{\text{program}}$$

$\mathcal{A}$: pontosítás, egyszerűsítés, absztrakció, lényegtelen elemek kiszűrése, a lényeg kihámozása

Modell: sokféle lehet, elég tág, de elég egyszerű, formalizált, pontos

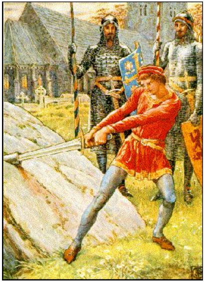
$\mathcal{B}$: hatékony algoritmus, bemenő adatok $\rightarrow$ eredmény, érdemes foglalkozni a kapott algoritmus *elemzésével, értékelésével*, megvizsgálva, hogy a módszer mennyire hatékony

# Arthur király civilizációs törekvései



Arthur király fényes udvarában 150 lovag és 150 udvarhölgy él. A király, aki közismert civilizációs erőfeszítéseiről, elhatározza, hogy megházasítja jó lovagjait és szép udvarhölgyeit. Mindezt persze emberségesen szeretné tenni. Csak olyan párok egybekelését akarja, amelyek tagjai kölcsönösen vonzalmat éreznek egymás iránt. Hogyan fogjon hozzá?

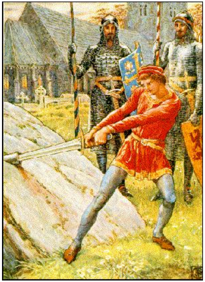
# Arthur király civilizációs törekvései



Arthur király fényes udvarában 150 lovag és 150 udvarhölgy él. A király, aki közismert civilizációs erőfeszítéseiről, elhatározza, hogy megházasítja jó lovagjait és szép udvarhölgyeit. Mindezt persze emberségesen szeretné tenni. Csak olyan párok egybekelését akarja, amelyek tagjai kölcsönösen vonzalmat éreznek egymás iránt. Hogyan fogjon hozzá?

Természetesen pártfogójához, a nagyhatalmú varázslóhoz, Merlinhez fordul. Merlin rögvest felismeri, hogy itt is bináris szimmetrikus viszonyok ábrázolásáról van szó.

# Arthur király civilizációs törekvései

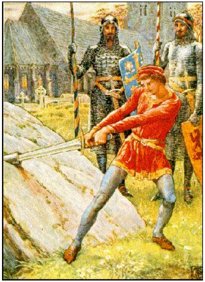


Arthur király fényes udvarában 150 lovag és 150 udvarhölgy él. A király, aki közismert civilizációs erőfeszítéseiről, elhatározza, hogy megházasítja jó lovagjait és szép udvarhölgyeit. Mindezt persze emberségesen szeretné tenni. Csak olyan párok egybekelését akarja, amelyek tagjai kölcsönösen vonzalmat éreznek egymás iránt. Hogyan fogjon hozzá?

Természetesen pártfogójához, a nagyhatalmú varázslóhoz, Merlinhez fordul. Merlin rögvest felismeri, hogy itt is bináris szimmetrikus viszonyok ábrázolásáról van szó.

Nagy darab pergament vesz elő, és nekilát egy páros gráfot rajzolni.

# Arthur király civilizációs törekvései

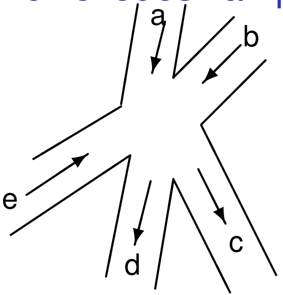


Arthur király fényes udvarában 150 lovag és 150 udvarhölgy él. A király, aki közismert civilizációs erőfeszítéseiről, elhatározza, hogy megházasítja jó lovagjait és szép udvarhölgyeit. Mindezt persze emberségesen szeretné tenni. Csak olyan párok egybekelését akarja, amelyek tagjai kölcsönösen vonzalmat éreznek egymás iránt. Hogyan fogjon hozzá?

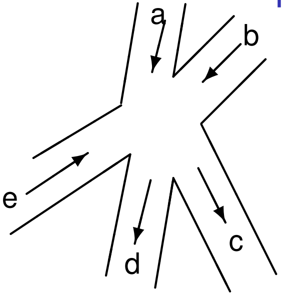
Természetesen pártfogójához, a nagyhatalmú varázslóhoz, Merlinhez fordul. Merlin rögvest felismeri, hogy itt is bináris szimmetrikus viszonyok ábrázolásáról van szó.

Nagy darab pergament vesz elő, és nekilát egy páros gráfot rajzolni. A királyi parancs teljesítéséhez Merlinnek élek egy olyan rendszerét kell kiválasztania a gráf éleiből, hogy a kiválasztott élek közül a gráf minden pontjához pontosan egy csatlakozzon. A kiválasztott élek felelnek meg a tervezett házasságoknak. A gráfelmélet nyelvén teljes párosítást kell keresnie.

# Közlekedési lámpák ütemezése



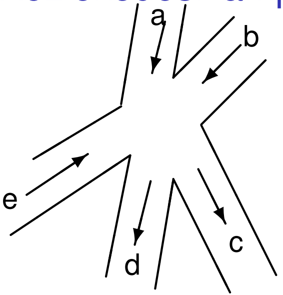
# Közlekedési lámpák ütemezése




lámpák: $ac, ad, bc, bd, ec$ és $ed$

állapot: lámpák $\rightarrow \{P, Z\}$

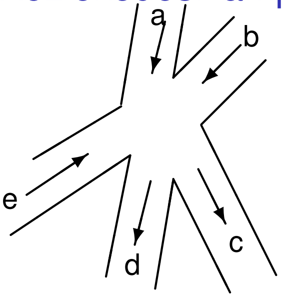
# Közlekedési lámpák ütemezése




lámpák: $ac, ad, bc, bd, ec$ és $ed$
állapot: lámpák $\rightarrow \{P, Z\}$

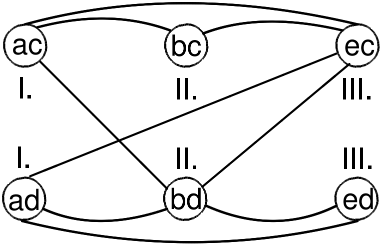
Feladat: Mennyi a minimális számú állapot, ami biztonságos és nem okoz örök dugót?

# Közlekedési lámpák ütemezése

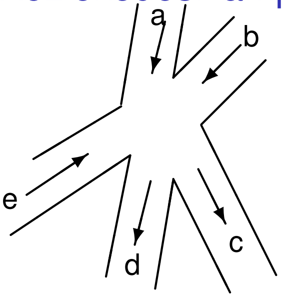



lámpák: $ac, ad, bc, bd, ec$ és $ed$
állapot: lámpák $\to \{P, Z\}$

Feladat: Mennyi a minimális számú állapot, ami biztonságos és nem okoz örök dugót?

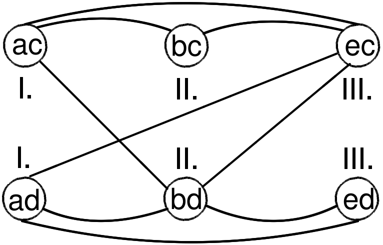


# Közlekedési lámpák ütemezése




lámpák: $ac, ad, bc, bd, ec$ és $ed$
állapot: lámpák $\to \{P, Z\}$

Feladat: Mennyi a minimális számú állapot, ami biztonságos és nem okoz örök dugót?



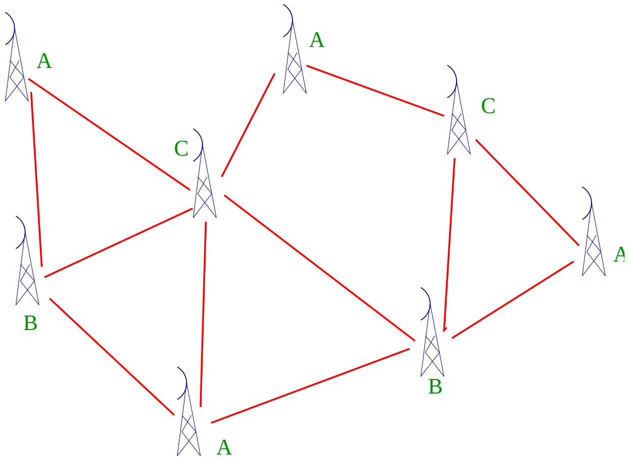

Gráfelméleti nyelven: Mennyi $G$ kromatikus száma?

# Mobiltelefon-átjátszók frekvencia kiosztása

Egy adott átjátszóhoz egy adott frenkvenciát rendelnek.
Egy telefon a közelben levő átjátszók közül választ.
„Közel levő” átjátszók frekvenciája különbözzön.

# Mobiltelefon-átjátszók frekvencia kiosztása

Egy adott átjátszóhoz egy adott frenkvenciát rendelnek.
Egy telefon a közelben levő átjátszók közül választ.
„Közel levő” átjátszók frekvenciája különbözzön.



# Elágazás és korlátozás

Legtöbbször van $c^n$-es algoritmus, de nem mindegy mekkora $c$.

# Elágazás és korlátozás

Legtöbbször van $c^n$-es algoritmus, de nem mindegy mekkora $c$.

Bontsunk esetekre, azokat alesetekre, $\ldots \Longrightarrow$ fa

# Elágazás és korlátozás

Legtöbbször van $c^n$-es algoritmus, de nem mindegy mekkora $c$.

Bontsunk esetekre, azokat alesetekre, ... $\Longrightarrow$ fa
Értékeljük az eseteket $\Longrightarrow$ bizonyos irányokba nem kell továbbmenni.
$\Longrightarrow$ (korlátozó heurisztika)

# Elágazás és korlátozás

Legtöbbször van $c^n$-es algoritmus, de nem mindegy mekkora $c$.

Bontsunk esetekre, azokat alesetekre, ... $\Longrightarrow$ fa
Értékeljük az eseteket $\Longrightarrow$ bizonyos irányokba nem kell továbbmenni.
$\Longrightarrow$ (korlátozó heurisztika)
Pl. sakkállások

# Elágazás és korlátozás

Legtöbbször van $c^n$-es algoritmus, de nem mindegy mekkora $c$.

Bontsunk esetekre, azokat alesetekre, ... $\Longrightarrow$ fa
Értékeljük az eseteket $\Longrightarrow$ bizonyos irányokba nem kell továbbmenni.
$\Longrightarrow$ (korlátozó heurisztika)
Pl. sakkállások

Feladat: Keressünk maximális méretű független ponthalmazt egy adott $G$ gráfban.

# Elágazás és korlátozás

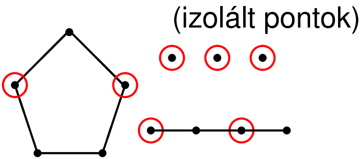
Legtöbbször van $c^n$-es algoritmus, de nem mindegy mekkora $c$.

Bontsunk esetekre, azokat alesetekre, ... $\Longrightarrow$ fa
Értékeljük az eseteket $\Longrightarrow$ bizonyos irányokba nem kell továbbmenni.
$\Longrightarrow$ (korlátozó heurisztika)
Pl. sakkállások

Feladat: Keressünk maximális méretű független ponthalmazt egy adott $G$ gráfban.
Nyilvánvaló módszer:
Minden részhalmazt végignézünk $\Longrightarrow$ $O(2^n)$ lépés

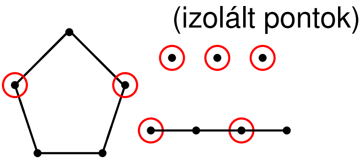
# Jobb algoritmus

**Észrevétel:** *Ha G-ben minden pont foka legfeljebb kettő, akkor a feladat lineáris időben megoldható: G izolált pontok, utak és körök diszjunkt uniója.* $\Longrightarrow$ *komponensenként minden „második" pontot bevesszük a halmazba.*



# Jobb algoritmus

**Észrevétel:** *Ha G-ben minden pont foka legfeljebb kettő, akkor a feladat lineáris időben megoldható: G izolált pontok, utak és körök diszjunkt uniója.* $\Longrightarrow$ *komponensenként minden „második" pont bevesszük a halmazba.*

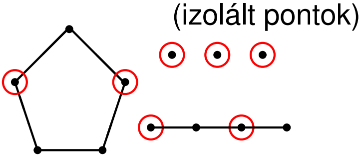



MF($G$)

1. Ha $G$-ben minden pont foka $\leq 2$, akkor MF($G$) az előbbi eljárás által adott maximális független halmaz, és a munkát befejeztük.

# Jobb algoritmus

**Észrevétel:** *Ha G-ben minden pont foka legfeljebb kettő, akkor a feladat lineáris időben megoldható: G izolált pontok, utak és körök diszjunkt uniója.* $\Longrightarrow$ *komponensenként minden „második" pontot bevesszük a halmazba.*



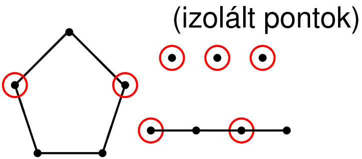

MF($G$)

1. Ha $G$-ben minden pont foka $\leq 2$, akkor MF($G$) az előbbi eljárás által adott maximális független halmaz, és a munkát befejeztük.
2. Legyen $x \in G$, $fok(x) \geq 3$.

   $S_1 :=$ MF$(G \setminus \{x\})$

# Jobb algoritmus

**Észrevétel:** *Ha G-ben minden pont foka legfeljebb kettő, akkor a feladat lineáris időben megoldható: G izolált pontok, utak és körök diszjunkt uniója.* $\Longrightarrow$ *komponensenként minden „második" pontot bevesszük a halmazba.*




MF($G$)

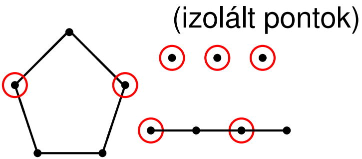
1. Ha $G$-ben minden pont foka $\leq 2$, akkor MF($G$) az előbbi eljárás által adott maximális független halmaz, és a munkát befejeztük.
2. Legyen $x \in G$, $fok(x) \geq 3$.

   $S_1 := \text{MF}(G \setminus \{x\})$

   $S_2 := \{x\} \cup \text{MF}(G \setminus \{x \text{ és szomszédai}\})$.

# Jobb algoritmus

**Észrevétel:** *Ha G-ben minden pont foka legfeljebb kettő, akkor a feladat lineáris időben megoldható: G izolált pontok, utak és körök diszjunkt uniója.* $\Longrightarrow$ *komponensenként minden „második" pontot bevesszük a halmazba.*

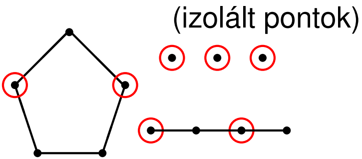



MF($G$)

1. Ha $G$-ben minden pont foka $\leq 2$, akkor MF($G$) az előbbi eljárás által adott maximális független halmaz, és a munkát befejeztük.
2. Legyen $x \in G$, $fok(x) \geq 3$.
   $S_1 := \text{MF}(G \setminus \{x\})$
   $S_2 := \{x\} \cup \text{MF}(G \setminus \{x \text{ és szomszédai}\})$.
3. Legyen $S$ az $S_1$ és $S_2$ közül a nagyobb méretű, illetve akármelyik, ha $|S_1| = |S_2|$.

# Jobb algoritmus

**Észrevétel:** *Ha G-ben minden pont foka legfeljebb kettő, akkor a feladat lineáris időben megoldható: G izolált pontok, utak és körök diszjunkt uniója.* $\Longrightarrow$ *komponensenként minden „második" pont bevesszük a halmazba.*




MF($G$)

1. Ha $G$-ben minden pont foka $\leq 2$, akkor MF($G$) az előbbi eljárás által adott maximális független halmaz, és a munkát befejeztük.
2. Legyen $x \in G$, $fok(x) \geq 3$.

   $S_1 := \text{MF}(G \setminus \{x\})$

   $S_2 := \{x\} \cup \text{MF}(G \setminus \{x \text{ és szomszédai}\})$.
3. Legyen $S$ az $S_1$ és $S_2$ közül a nagyobb méretű, illetve akármelyik, ha $|S_1| = |S_2|$.
4. MF($G$) $:= S$.

Legyen $T(n)$ az MF($G$)-n ($|V(G)| \leq n$) belüli MF hívások maximális száma, beleértve MF($G$)-t magát is.

Legyen $T(n)$ az MF($G$)-n ($|V(G)| \leq n$) belüli MF hívások maximális száma, beleértve MF($G$)-t magát is.

**Tétel**

*Van olyan c állandó, hogy $T(n) \leq c\gamma^n$, tetszőleges n természetes számra, ahol $\gamma$ a $\gamma^4 - \gamma^3 - 1 = 0$ egyenlet pozitív gyöke ($\gamma \approx 1,381$).*

Legyen $T(n)$ az MF($G$)-n ($|V(G)| \leq n$) belüli MF hívások maximális száma, beleértve MF($G$)-t magát is.

## Tétel

*Van olyan c állandó, hogy $T(n) \leq c\gamma^n$, tetszőleges n természetes számra, ahol $\gamma$ a $\gamma^4 - \gamma^3 - 1 = 0$ egyenlet pozitív gyöke ($\gamma \approx 1,381$).*

## Bizonyítás.

Legyen $t(n) := T(n) + 1$.

Legyen $T(n)$ az MF($G$)-n ($|V(G)| \leq n$) belüli MF hívások maximális száma, beleértve MF($G$)-t magát is.

## Tétel

*Van olyan c állandó, hogy $T(n) \leq c\gamma^n$, tetszőleges n természetes számra, ahol $\gamma$ a $\gamma^4 - \gamma^3 - 1 = 0$ egyenlet pozitív gyöke ($\gamma \approx 1,381$).*

## Bizonyítás.

Legyen $t(n) := T(n) + 1$.
$T(n) \leq T(n-1) + T(n-4) + 1$, ha $n > 4$.

Legyen $T(n)$ az MF($G$)-n ($|V(G)| \leq n$) belüli MF hívások maximális száma, beleértve MF($G$)-t magát is.

## Tétel

*Van olyan c állandó, hogy $T(n) \leq c\gamma^n$, tetszőleges n természetes számra, ahol $\gamma$ a $\gamma^4 - \gamma^3 - 1 = 0$ egyenlet pozitív gyöke ($\gamma \approx 1,381$).*

## Bizonyítás.

Legyen $t(n) := T(n) + 1$.
$T(n) \leq T(n-1) + T(n-4) + 1$, ha $n > 4$. $\Longrightarrow$
$t(n) \leq t(n-1) + t(n-4)$, ha $n > 4$.

Legyen $T(n)$ az MF($G$)-n ($|V(G)| \leq n$) belüli MF hívások maximális száma, beleértve MF($G$)-t magát is.

## Tétel

*Van olyan c állandó, hogy $T(n) \leq c\gamma^n$, tetszőleges n természetes számra, ahol $\gamma$ a $\gamma^4 - \gamma^3 - 1 = 0$ egyenlet pozitív gyöke ($\gamma \approx 1,381$).*

## Bizonyítás.

Legyen $t(n) := T(n) + 1$.
$T(n) \leq T(n-1) + T(n-4) + 1$, ha $n > 4$. $\Longrightarrow$
$t(n) \leq t(n-1) + t(n-4)$, ha $n > 4$.
Indukcióval: $t(n) \leq c\gamma^n$ igaz

Legyen $T(n)$ az MF($G$)-n ($|V(G)| \leq n$) belüli MF hívások maximális száma, beleértve MF($G$)-t magát is.

## Tétel

*Van olyan c állandó, hogy $T(n) \leq c\gamma^n$, tetszőleges n természetes számra, ahol $\gamma$ a $\gamma^4 - \gamma^3 - 1 = 0$ egyenlet pozitív gyöke ($\gamma \approx 1,381$).*

## Bizonyítás.

Legyen $t(n) := T(n) + 1$.
$T(n) \leq T(n-1) + T(n-4) + 1$, ha $n > 4$. $\implies$
$t(n) \leq t(n-1) + t(n-4)$, ha $n > 4$.
Indukcióval: $t(n) \leq c\gamma^n$ igaz$n < 5$-re elég nagy $c$-vel $\surd$

Legyen $T(n)$ az MF($G$)-n ($|V(G)| \leq n$) belüli MF hívások maximális száma, beleértve MF($G$)-t magát is.

## Tétel

*Van olyan c állandó, hogy $T(n) \leq c\gamma^n$, tetszőleges n természetes számra, ahol $\gamma$ a $\gamma^4 - \gamma^3 - 1 = 0$ egyenlet pozitív gyöke ($\gamma \approx 1,381$).*

## Bizonyítás.

Legyen $t(n) := T(n) + 1$.
$T(n) \leq T(n-1) + T(n-4) + 1$, ha $n > 4$. $\Longrightarrow$
$t(n) \leq t(n-1) + t(n-4)$, ha $n > 4$.
Indukcióval: $t(n) \leq c\gamma^n$ igaz $n < 5$-re elég nagy $c$-vel $\surd$
$\Longrightarrow$ Ezután, ha $n \geq 5$, indukciós feltevésből:

$$\begin{aligned} t(n) &\leq t(n-1) + t(n-4) \leq c\gamma^{n-1} + c\gamma^{n-4} = \\ &= c\gamma^{n-4}(\gamma^3 + 1) = c\gamma^{n-4}\gamma^4 = c\gamma^n. \end{aligned}$$

$\square$

Összköltség: $O(n^d T(n)) = O(n^d \gamma^n) = O(1,381^n)$.

# 3-színezés keresése

Feladat: Adott $G$, keressünk egy 3-színezést.

# 3-színezés keresése

Feladat: Adott $G$, keressünk egy 3-színezést.

Minden lehetséges színezést végignézünk $\Longrightarrow$ $O(3^n)$ lépés

# 3-színezés keresése

Feladat: Adott $G$, keressünk egy 3-színezést.

Minden lehetséges színezést végignézünk $\Longrightarrow$ $O(3^n)$ lépés

Ötlet: Bizonyos csúcsokat kiszínezünk pirosra, a többiről polinom időben el tudjuk dönteni, hogy kiszínezhetők-e kékkel és sárgával.

# 3-színezés keresése

Feladat: Adott $G$, keressünk egy 3-színezést.

Minden lehetséges színezést végignézünk $\Longrightarrow$ $O(3^n)$ lépés

Ötlet: Bizonyos csúcsokat kiszínezünk pirosra, a többiről polinom időben el tudjuk dönteni, hogy kiszínezhetők-e kékkel és sárgával.

Összköltség: $O(2^n n^c)$.

# Dinamikus programozás

Optimum meghatározása kisebb részfeladatok optimumainak felhasználásával.

# Dinamikus programozás

Optimum meghatározása kisebb részfeladatok optimumainak felhasználásával.

Általában egy táblázat kitöltése, az új elemeket a korábban kitöltött elemekből számoljuk.

## Binomiális együtthatók kiszámítása

$$
\begin{array}{ccccccccc}
 & & & & \binom{0}{0} & & & & \\
 & & & \binom{1}{0} & & \binom{1}{1} & & & \\
 & & \binom{2}{0} & & \binom{2}{1} & & \binom{2}{2} & & \\
 & \binom{3}{0} & & \binom{3}{1} & & \binom{3}{2} & & \binom{3}{3} & \\
\binom{4}{0} & & \binom{4}{1} & & \binom{4}{2} & & \binom{4}{3} & & \binom{4}{4}
\end{array}
\qquad
\begin{array}{ccccc}
\binom{0}{0} & & & & \\
\binom{1}{0} & \binom{1}{1} & & & \\
\binom{2}{0} & \binom{2}{1} & \binom{2}{2} & & \\
\binom{3}{0} & \binom{3}{1} & \binom{3}{2} & \binom{3}{3} & \\
\binom{4}{0} & \binom{4}{1} & \binom{4}{2} & \binom{4}{3} & \binom{4}{4}
\end{array}
$$

# Dinamikus programozás

Optimum meghatározása kisebb részfeladatok optimumainak felhasználásával.

Általában egy táblázat kitöltése, az új elemeket a korábban kitöltött elemekből számoljuk.

## Binomiális együtthatók kiszámítása

$$
\begin{array}{ccccccccc}
 & & & & \binom{0}{0} & & & & \\
 & & & \binom{1}{0} & & \binom{1}{1} & & & \\
 & & \binom{2}{0} & & \binom{2}{1} & & \binom{2}{2} & & \\
 & \binom{3}{0} & & \binom{3}{1} & & \binom{3}{2} & & \binom{3}{3} & \\
\binom{4}{0} & & \binom{4}{1} & & \binom{4}{2} & & \binom{4}{3} & & \binom{4}{4}
\end{array}
\qquad
\begin{array}{ccccc}
\binom{0}{0} & & & & \\
\binom{1}{0} & \binom{1}{1} & & & \\
\binom{2}{0} & \binom{2}{1} & \binom{2}{2} & & \\
\binom{3}{0} & \binom{3}{1} & \binom{3}{2} & \binom{3}{3} & \\
\binom{4}{0} & \binom{4}{1} & \binom{4}{2} & \binom{4}{3} & \binom{4}{4}
\end{array}
$$

### Tétel

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

# Hátizsák probléma

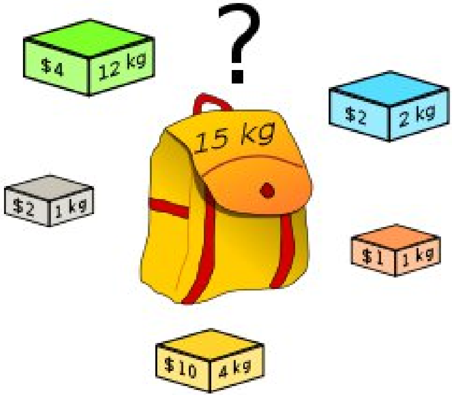
## Probléma

*Adottak az* $s_1, \ldots, s_m$ *súlyok, a* $b$ *súlykorlát és a* $v_1, \ldots, v_m$ *értékek. (Minden érték pozitív egész.) Melyik az az* $I \subseteq \{1, \ldots, m\}$ *részhalmaz, melyre teljesül, hogy* $\sum_{i \in I} s_i \leq b$ *és* $\sum_{i \in I} v_i$ *maximális?*

# Hátizsák probléma

## Probléma

*Adottak az $s_1, \ldots, s_m$ súlyok, a $b$ súlykorlát és a $v_1, \ldots, v_m$ értékek. (Minden érték pozitív egész.) Melyik az az $I \subseteq \{1, \ldots, m\}$ részhalmaz, melyre teljesül, hogy $\sum_{i \in I} s_i \leq b$ és $\sum_{i \in I} v_i$ maximális?*



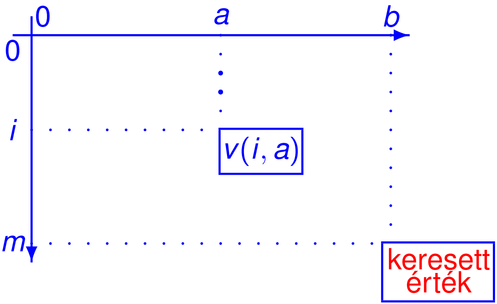
Először kisebb problémára oldjuk meg: $v(i, a)$ a maximális elérhető érték az $s_1, \ldots, s_i$ súlyokkal, $v_1, \ldots, v_i$ értékekkel és $a$ súlykorláttal megadott feladatra.

Először kisebb problémára oldjuk meg: $v(i, a)$ a maximális elérhető érték az $s_1, \dots, s_i$ súlyokkal, $v_1, \dots, v_i$ értékekkel és $a$ súlykorláttal megadott feladatra.
Ekkor $v(0, a) = v(i, 0) = 0$ $\forall a, i$-re

Először kisebb problémára oldjuk meg: $v(i, a)$ a maximális elérhető érték az $s_1, \ldots, s_i$ súlyokkal, $v_1, \ldots, v_i$ értékekkel és $a$ súlykorláttal megadott feladatra.

Ekkor $v(0, a) = v(i, 0) = 0$ $\forall a, i$-re

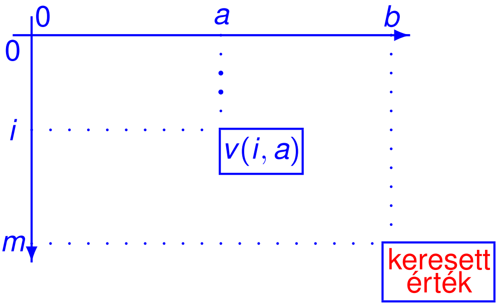
cél $\rightarrow v(m, b)$



Először kisebb problémára oldjuk meg: $v(i, a)$ a maximális elérhető érték az $s_1, \ldots, s_i$ súlyokkal, $v_1, \ldots, v_i$ értékekkel és $a$ súlykorláttal megadott feladatra.

Ekkor $v(0, a) = v(i, 0) = 0 \ \forall a, i$-re

cél $\to v(m, b)$

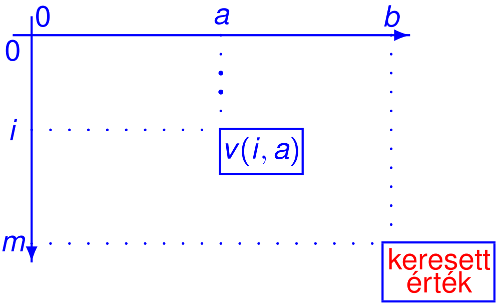



$$v(i, a) = \max\{v(i-1, a); v_i + v(i-1, a-s_i)\}$$

Először kisebb problémára oldjuk meg: $v(i,a)$ a maximális elérhető érték az $s_1,\dots,s_i$ súlyokkal, $v_1,\dots,v_i$ értékekkel és $a$ súlykorláttal megadott feladatra.

Ekkor $v(0,a) = v(i,0) = 0 \ \forall a, i$-re

cél $\rightarrow v(m,b)$



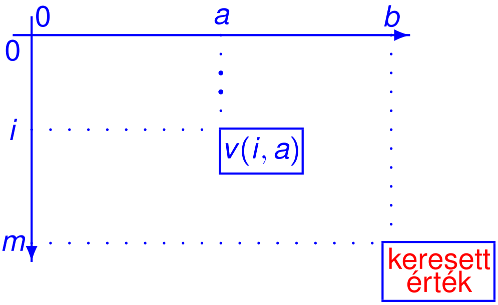

$$v(i,a) = \max\{v(i-1,a); v_i + v(i-1,a-s_i)\}$$

$\Longrightarrow$ Soronként kitölthető $\Longleftarrow$ minden érték két felette levőből számolható.

Először kisebb problémára oldjuk meg: $v(i, a)$ a maximális elérhető érték az $s_1, \dots, s_i$ súlyokkal, $v_1, \dots, v_i$ értékekkel és $a$ súlykorláttal megadott feladatra.

Ekkor $v(0, a) = v(i, 0) = 0 \ \forall a, i$-re

cél $\rightarrow v(m, b)$

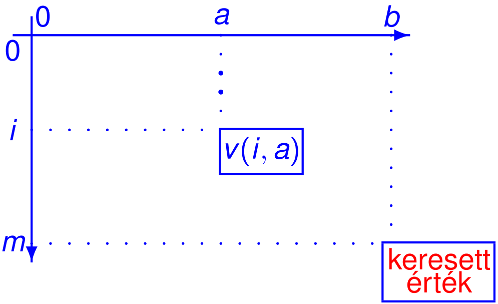



$$v(i, a) = \max\{v(i-1, a); v_i + v(i-1, a-s_i)\}$$

$\Longrightarrow$ Soronként kitölthető $\Longleftarrow$ minden érték két felette levőből számolható.

Összköltség: $O(bL)$

Először kisebb problémára oldjuk meg: $v(i, a)$ a maximális elérhető érték az $s_1, \dots, s_i$ súlyokkal, $v_1, \dots, v_i$ értékekkel és $a$ súlykorláttal megadott feladatra.
Ekkor $v(0, a) = v(i, 0) = 0$ $\forall a, i$-re
cél $\rightarrow v(m, b)$




$$v(i, a) = \max\{v(i-1, a); v_i + v(i-1, a-s_i)\}$$

$\Longrightarrow$ Soronként kitölthető $\Longleftarrow$ minden érték két felette levőből számolható.
Összköltség: $O(bL)$
$b$-től függ (nem $\log b$-től!), ha $b$ sokjegyű szám, ez sok idő