

Sample Solutions

CTU Open Contest 2011

ANALOG CLOCK



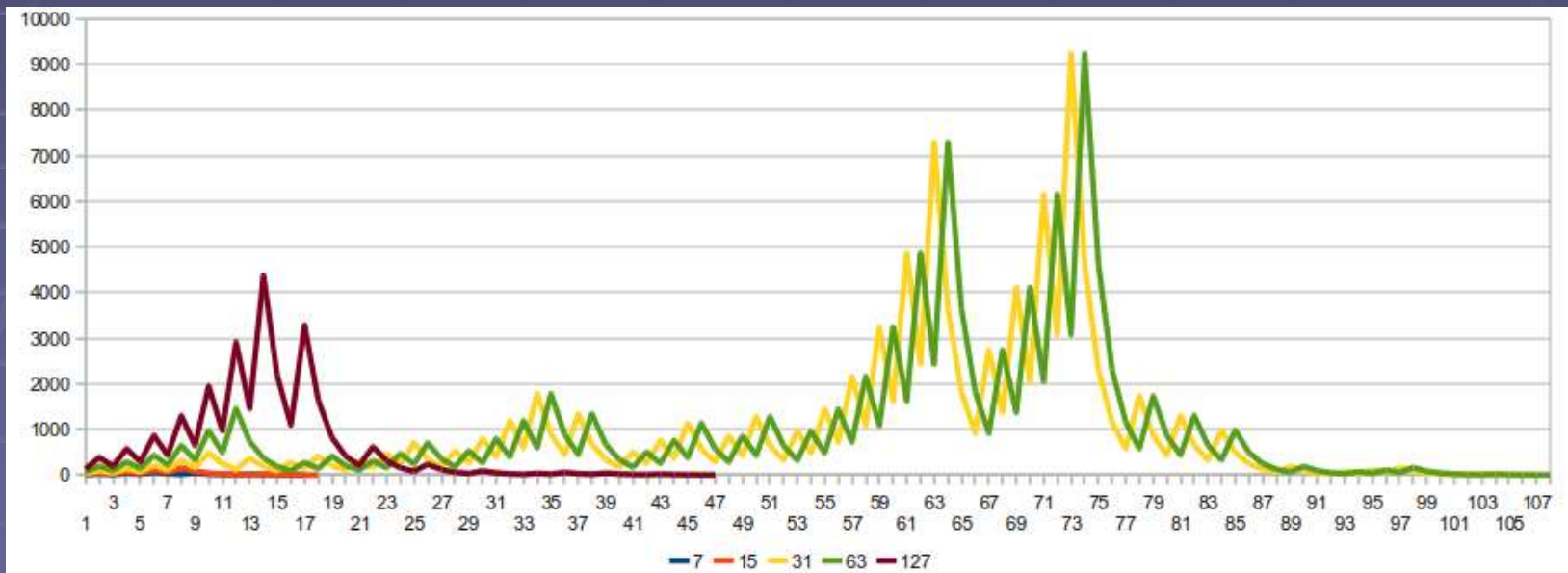
Analog Clock

- Prostě to “jen” poslechnout a naimplementovat...



- Úhel malé ručičky: $30^\circ \cdot H + 0,5^\circ \cdot M$
- Úhel velké ručičky: $6^\circ \cdot M$

COLLATZ CONJ.



Collatz Conjecture

- Stačilo nasimulovat a hledat
- Jak poznat první společné číslo?
 - Hledáme v mapě
 - Nebo si všimneme, že konec je vždy stejný => dojdeme na 1 a „couváme“, dokud jsou čísla stejná

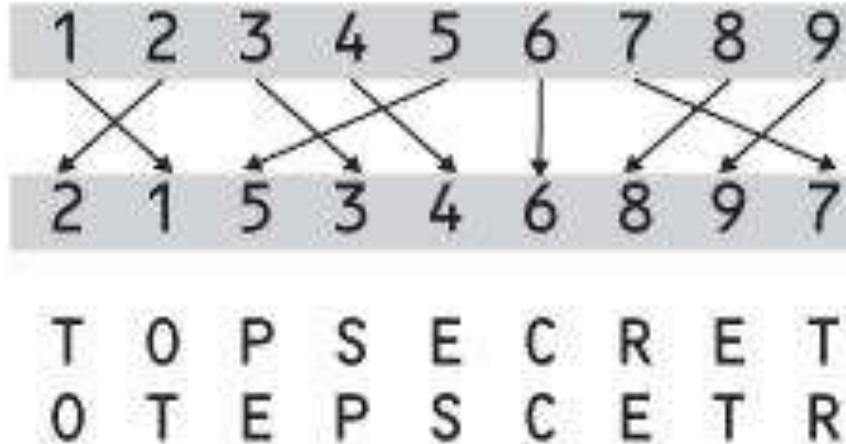
Collatz Conjecture

- Měla to být jednoduchá úloha...
- ... ale Collatzova posloupnost je záludná

- V čem byl teda problém?
 - $837799 \rightarrow \dots \rightarrow 2974984576 \rightarrow \dots$
 - NESTAČÍ NÁM INTEGER!!! (\Rightarrow long)

ENCRYPTION

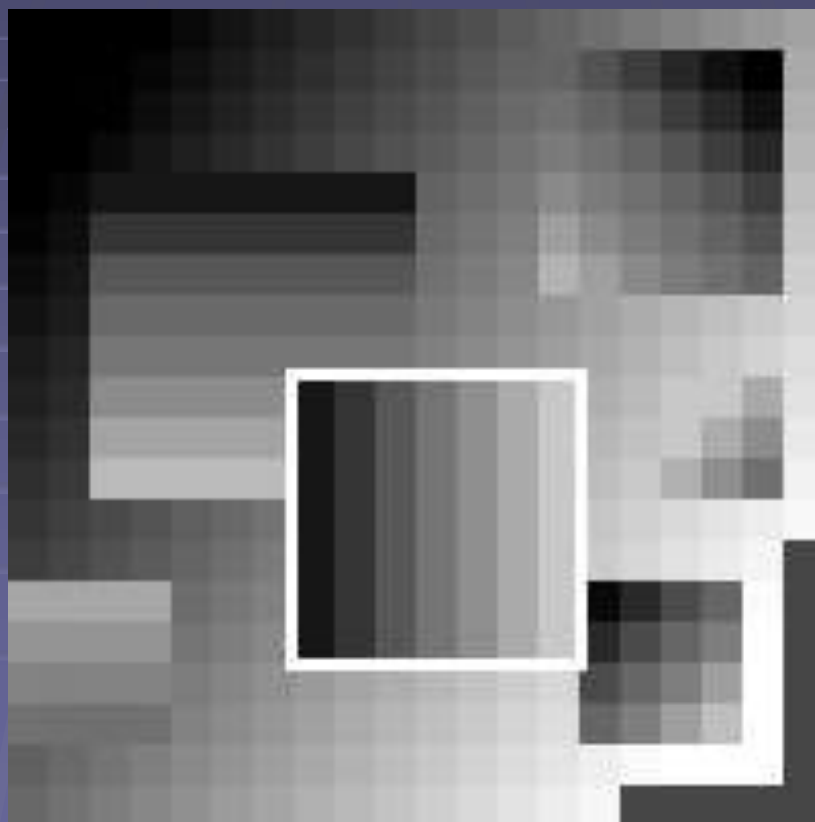
④ Transposition Cipher



Encryption

- Poměrně jednoduchá úloha ...
- ... mírně zkomplikovaná „vylepšením“ zadání na poslední chvíli

GRADIENT



Mine the Gradient

- Největší čtverec vyplněný gradientem
- Budeme zkoušet všechny?
To asi ne...
- Dynamické programování

Mine the Gradient

- Pro každé políčko spočítáme údaj:
 - Jaký největší čtverec v něm končí?
- To zjistíme v konstantním čase
 - Podle políček vlevo a nahoře
 - Plus protilehlý roh

Mine the Gradient – kód

```
max[i][j] = checkSmall(dir, i, j);
if (max[i][j] > best)
    best = max[i][j];

int m = Math.min(max[i-1][j], max[i][j-1]);
if (m < 3) continue;
int d = pic[i-1][j-1] - pic[i-2][j-2];
if (pic[i][j] - pic[i-1][j-1] != d) continue;
max[i][j] =
    (pic[i-m][j-m] + m*d == pic[i][j])
    ? m+1 : m;
if (max[i][j] > best)
    best = max[i][j];
```


INVASION



Invasion

- Graf
- Hledáme vrcholy s nejkratší vzdáleností menší než K
- \Rightarrow Dijkstraův algoritmus

Invasion

- Dijkstraův algoritmus – inicializace
 - Na začátku všechna města $+\infty$ (nebo K)
- Pro každou novou základnu
 - Nastavím její ohodnocení na 0
 - Pustím algoritmus (relaxaci), dokud se nezastaví
 - ... atd.

Invasion

- Efektivita
 - Stačí toto jednoduché řešení
- Vrcholy se vzdáleností $< K$
 - Neměli bychom je po každé základně počítat znovu
 - Pamatujeme si počet a aktualizujeme

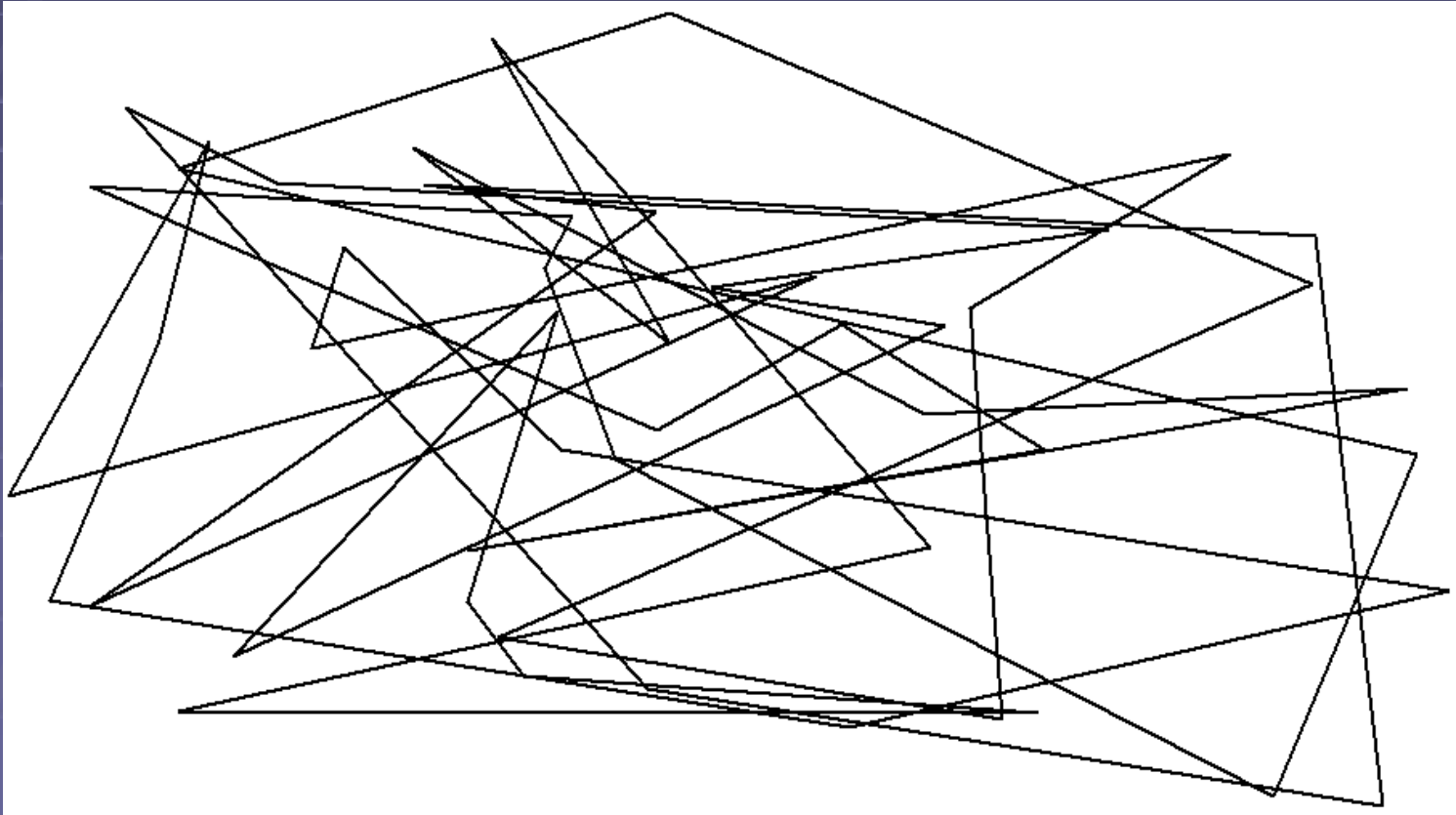
MORTGAGE



Intergalactic Mortgage

- Udržujeme stav dluhu (B)
- Na konci každého měsíce
 - Přičteme úrok: $B = B \cdot (1 + r/12)$
 - Odečteme splátku: $B = B - Y$
- Ale smyčkou se to nestihne...

POLYGON



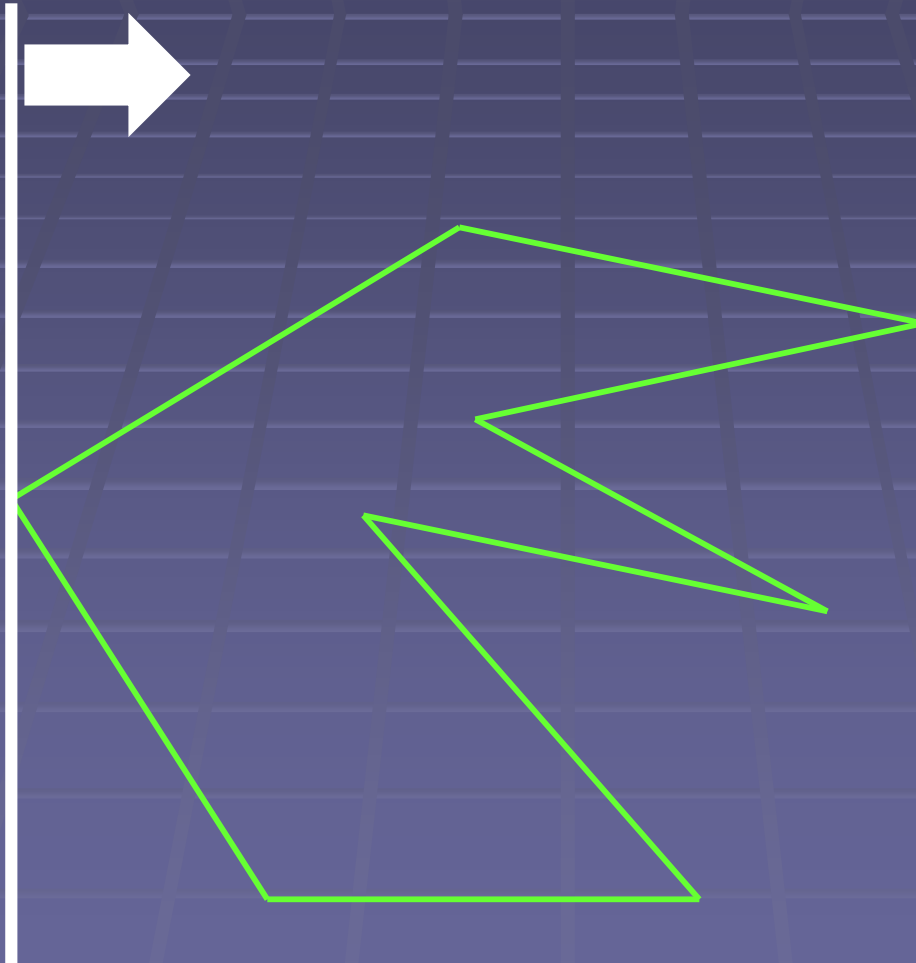
Simple Polygon

- Kontrolujeme průsečík dvojic úseček
- Ale opět nikoli každá s každou...
- => „Zametací přímka“

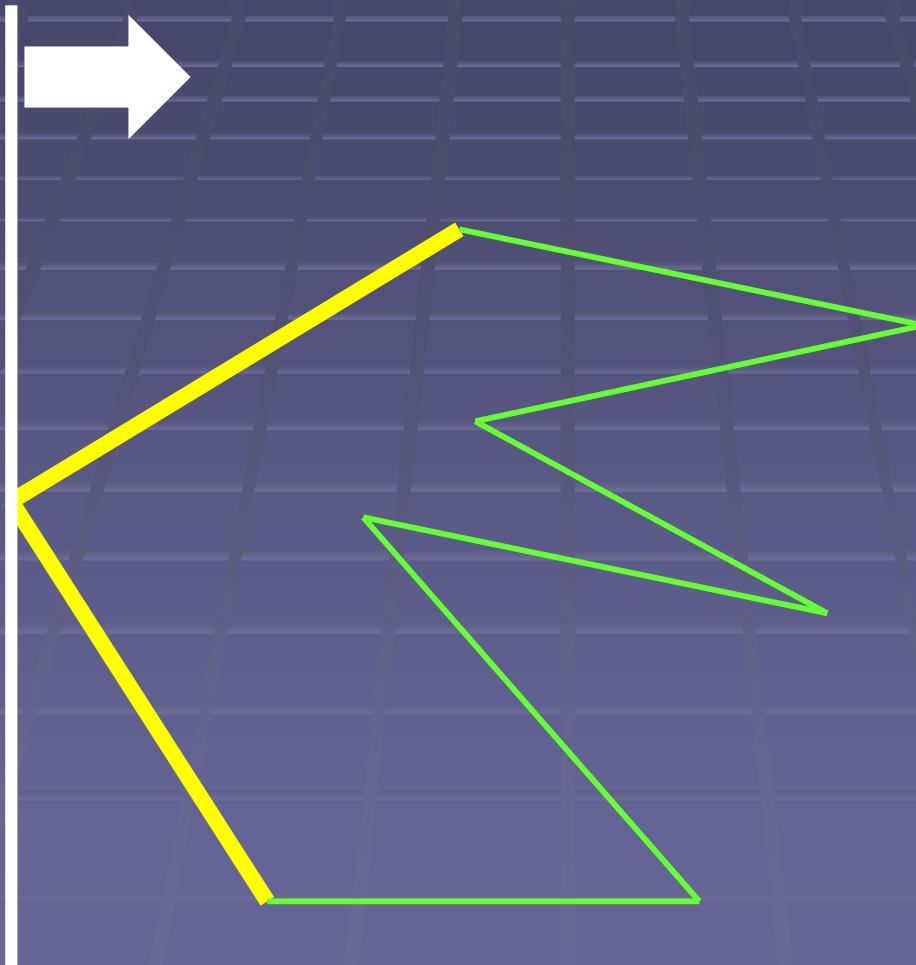
Simple Polygon

- Zametací přímka
- Udržujeme seřazený seznam úseček
 - protínajících přímku
 - aktualizujeme ($\log N$)
- => Složitost $O(N \cdot \log N)$

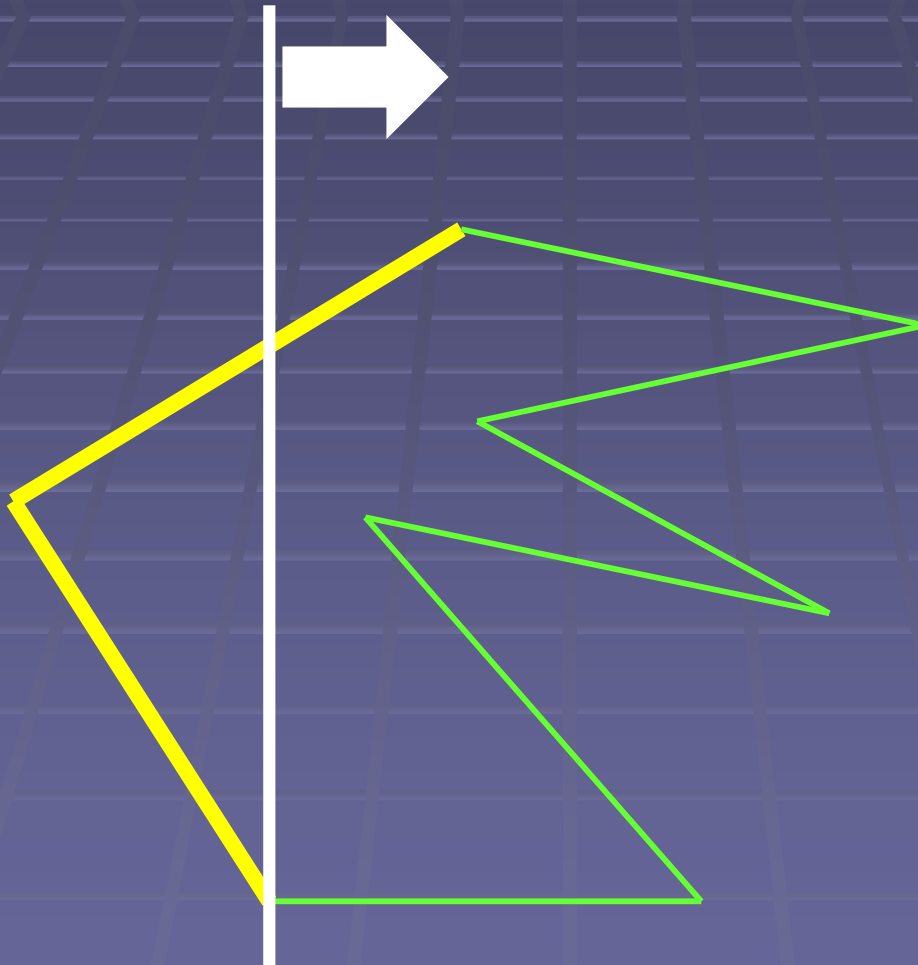
Simple Polygon



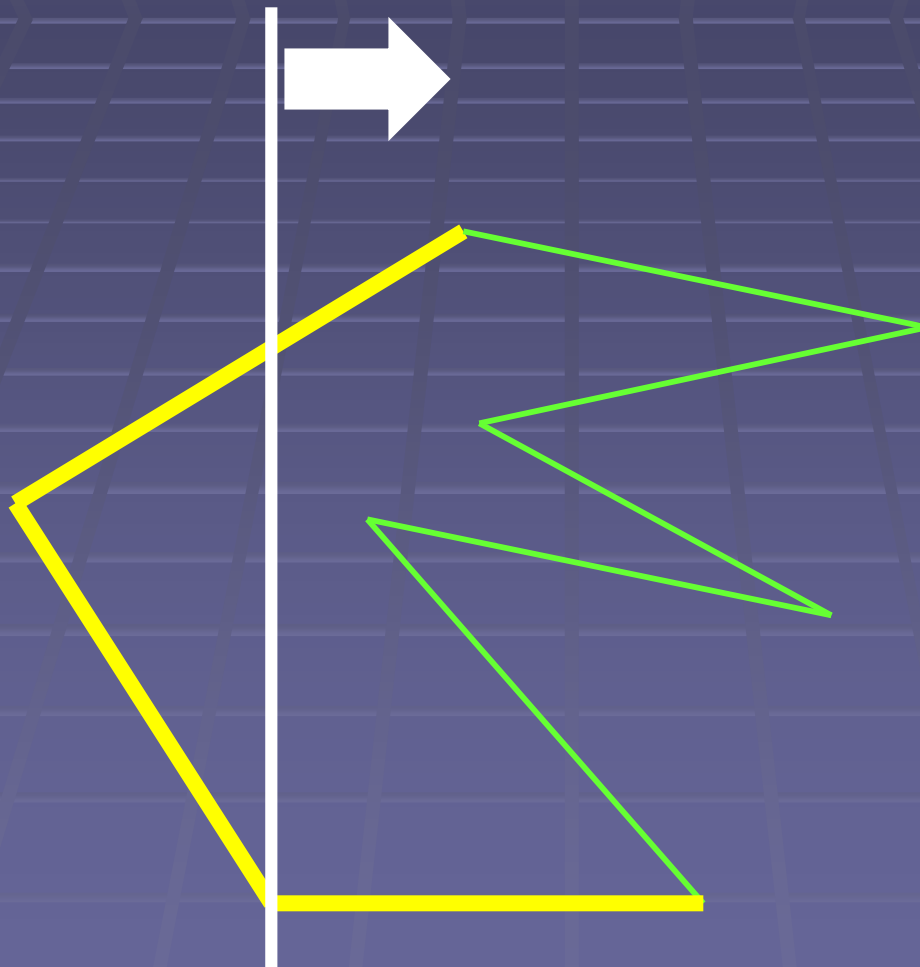
Simple Polygon



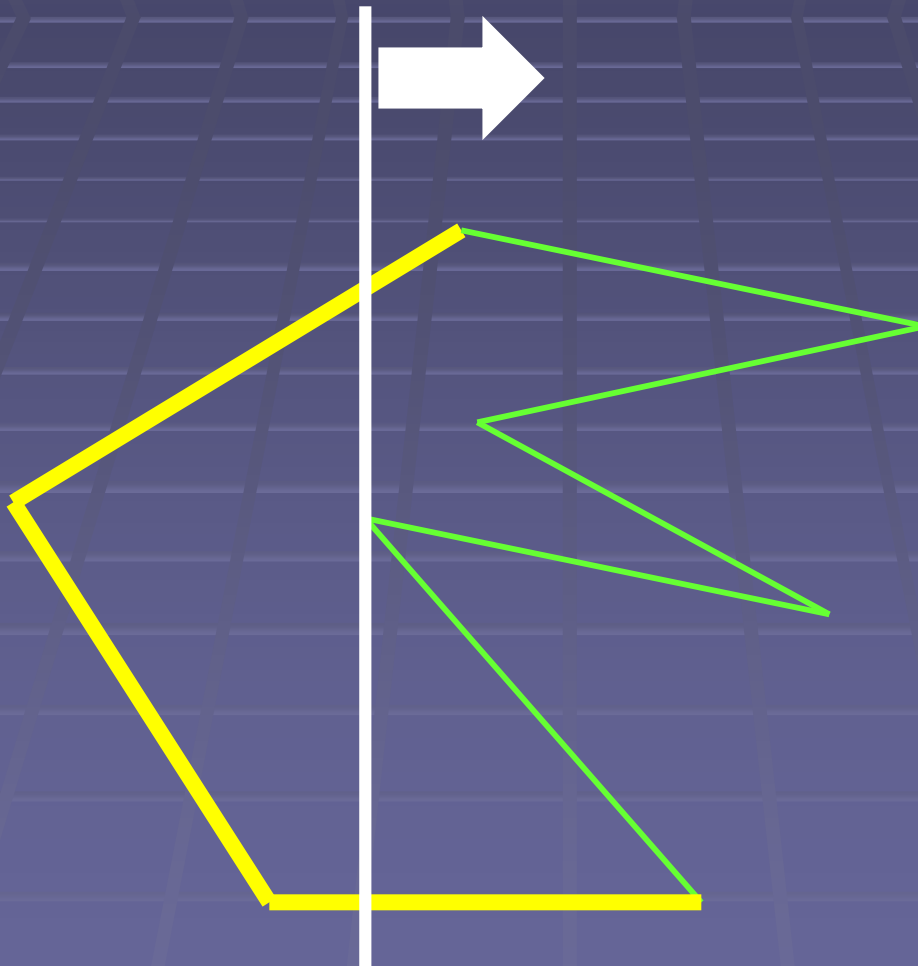
Simple Polygon



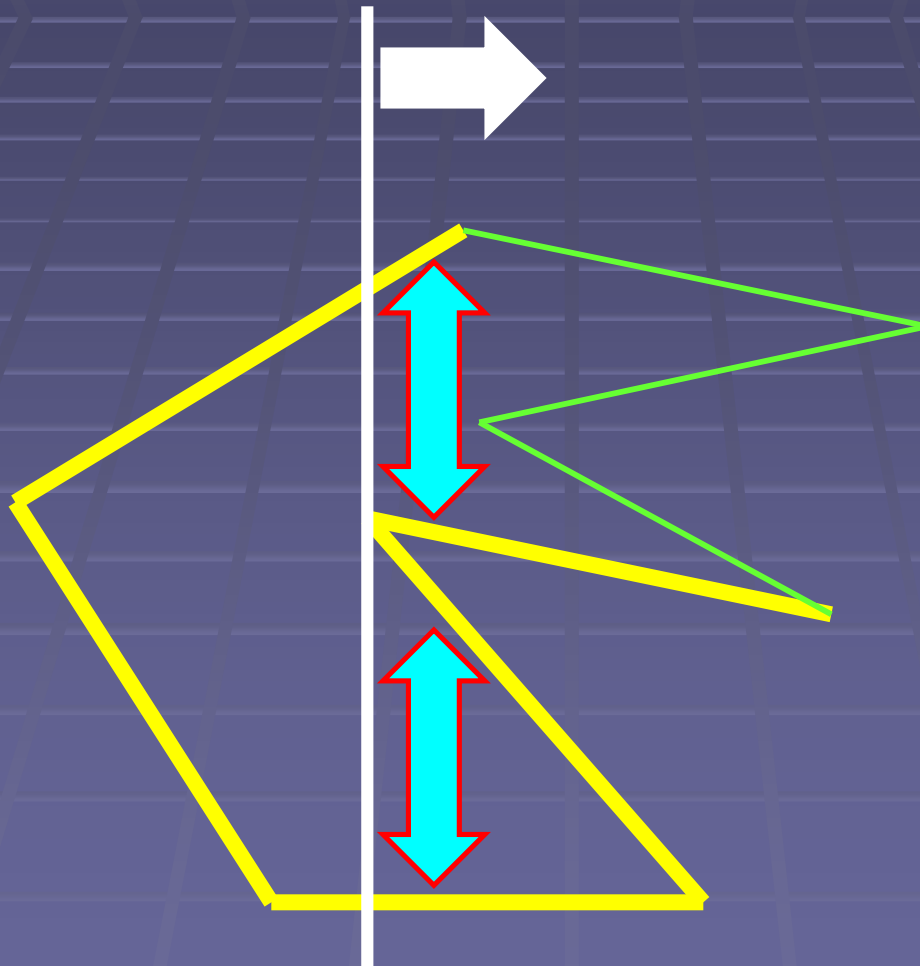
Simple Polygon



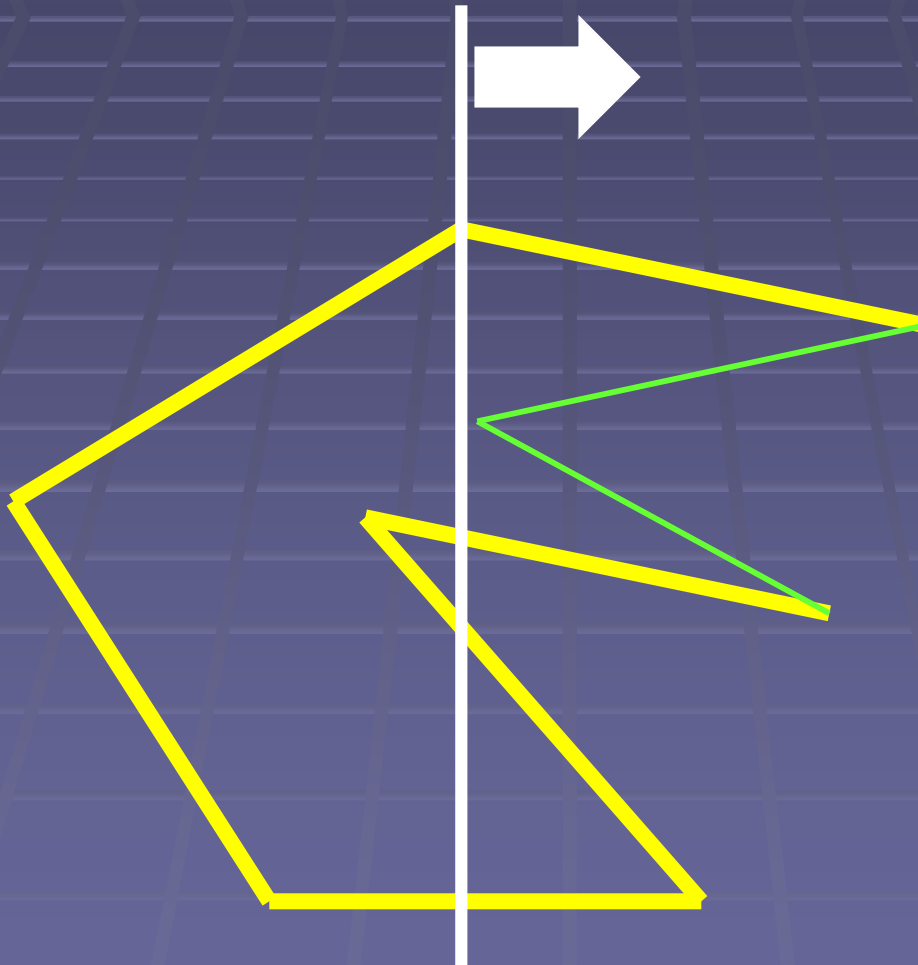
Simple Polygon



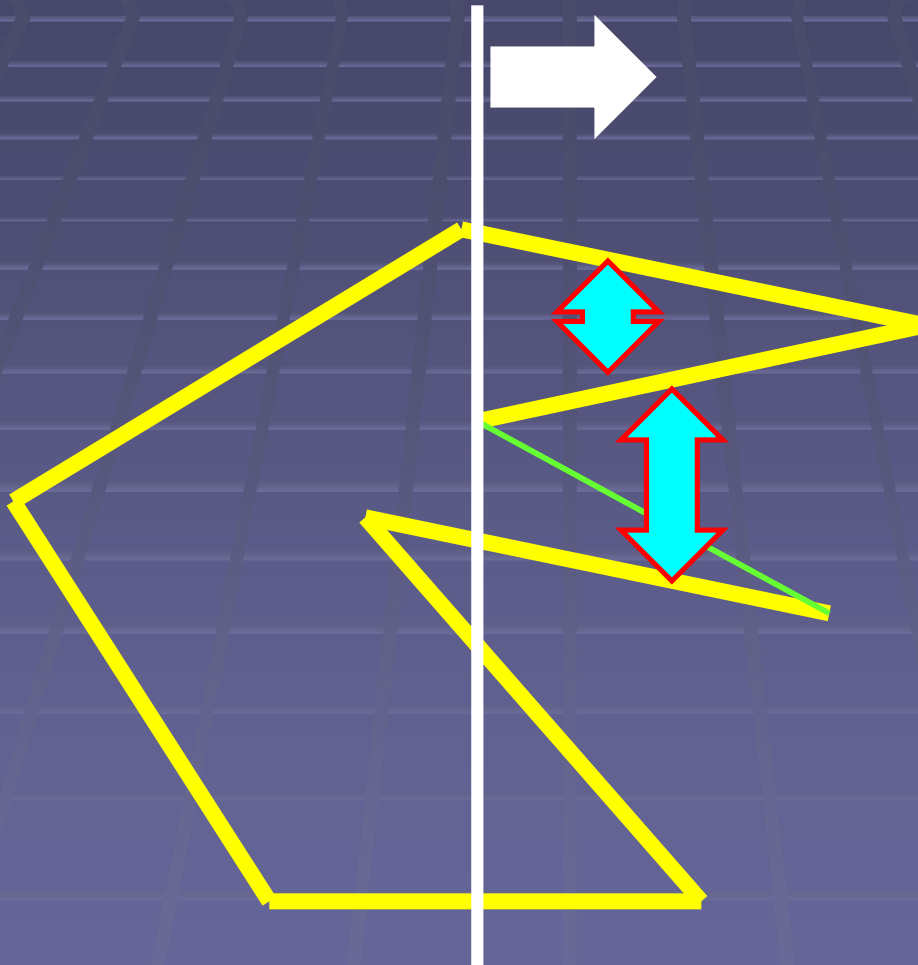
Simple Polygon



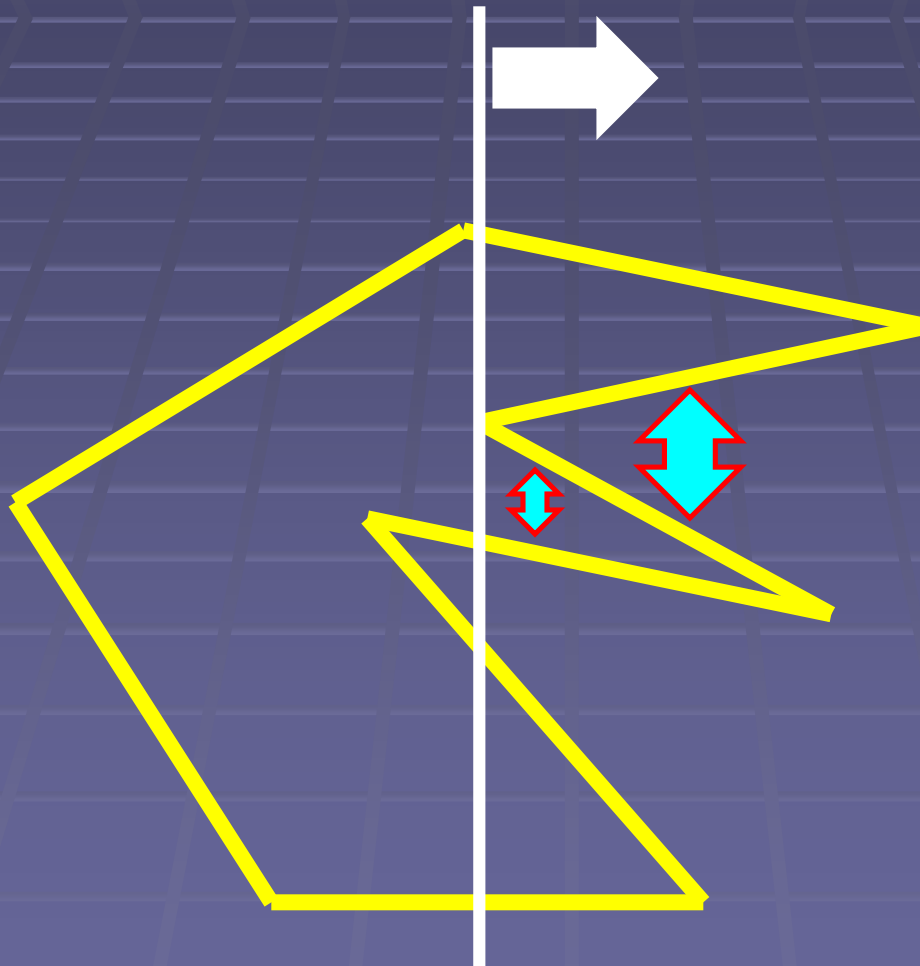
Simple Polygon



Simple Polygon



Simple Polygon



RESULT

1 + 4 * 7 * 2 + 5 * 8 + 9 * 3

Ambiguous Result

- Dynamické programování
- Vyzkoušíme nejlepší řešení pro každou část výrazu (od–do)
 - Postupně od kratších k delším

Ambiguous Result

- Jak zjistím nejlepší výsledek pro část výrazu I až J ?
 - Postupně zkusím rozdělovat:
 - (I až k) + (k až J)
 - (I až k) * (k až J)

1 + 4 * 7 * 2 + 5 * 8 + 9 * 3

Ambiguous Result

- Jak zjistím nejlepší výsledek pro část výrazu I až J ?
 - Postupně zkusím rozdělovat:
 - (I až k) + (k až J)
 - (I až k) * (k až J)

1 + 4 * 7 * 2 + 5 * 8 + 9 * 3

Ambiguous Result

- Jak zjistím nejlepší výsledek pro část výrazu I až J ?
 - Postupně zkusím rozdělovat:
 - (I až k) + (k až J)
 - (I až k) * (k až J)

1 + 4 * 7 * 2 + 5 * 8 + 9 * 3

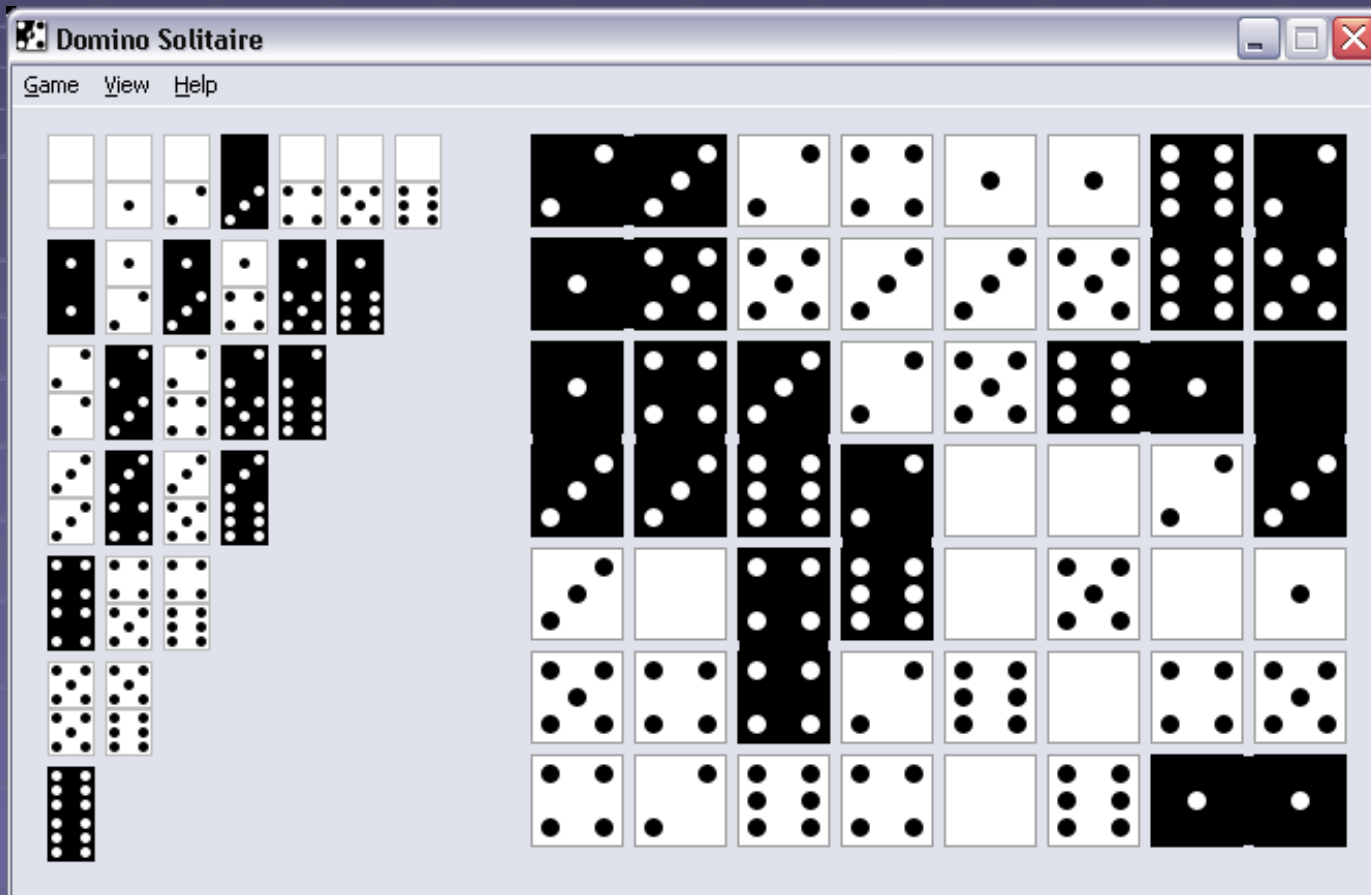
Ambiguous Result

- Ze všech možností rozdělení vybereme nejmenší a největší
- Pokud postupujeme od kratších intervalů k delším, tak ty menší už známe

Ambiguous Result

- Alternativní řešení
 - Maximum
 1. Násobení nulou
 2. Sčítání
 3. Ostatní násobení
 - Maximum
 - Analogicky

DOMINO TILING



Domino Tiling

- Prostě to zkusíme...
- Backtracking
 - Rozumné ořezávání
 - ... ale byli jsme hodní

Domino Tiling

- Jak nejlépe postupovat?
- Začnu tím, co má nejméně možností
 - Kostička, která má nejméně možností
 - Pole, na které už jde dát jen jedna

Domino Tiling

0	4	0	0	2	3
0	0	4	0	4	0
4	4	2	1	1	4
1	2	0	3	3	3
1	2	1	0	1	3

Domino Tiling

0	4	0	0	2	3
0	0	4	0	4	0
4	4	2	1	1	4
1	2	3	0	3	3
1	2	1	0	1	3

Domino Tiling

0	4	0	0	2	3
0	0	4	0	4	0
4	4	2	1	1	4
1	2	3	0	3	3
1	2	1	0	1	3

Autoři úloh

Josef Cibulka

Jakub Černý

Zdeněk Dvořák

Martin Kačer

Jan Stoklasa

Marko Berezovský

Jan Katrenic

Pavel Kos

Radek Pelánek