

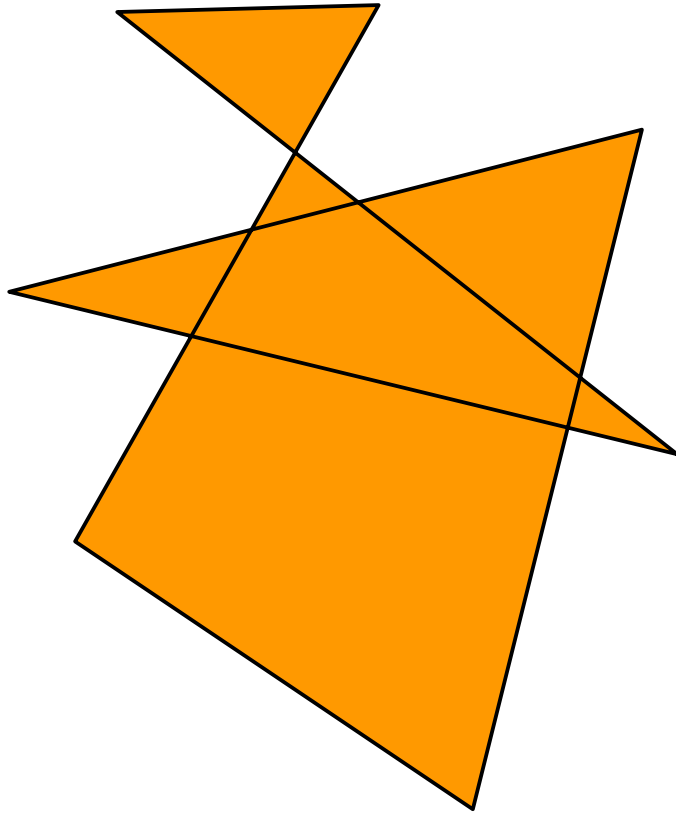
Vyplňování n-úhelníka

© 1995-2019 Josef Pelikán
CGG MFF UK Praha

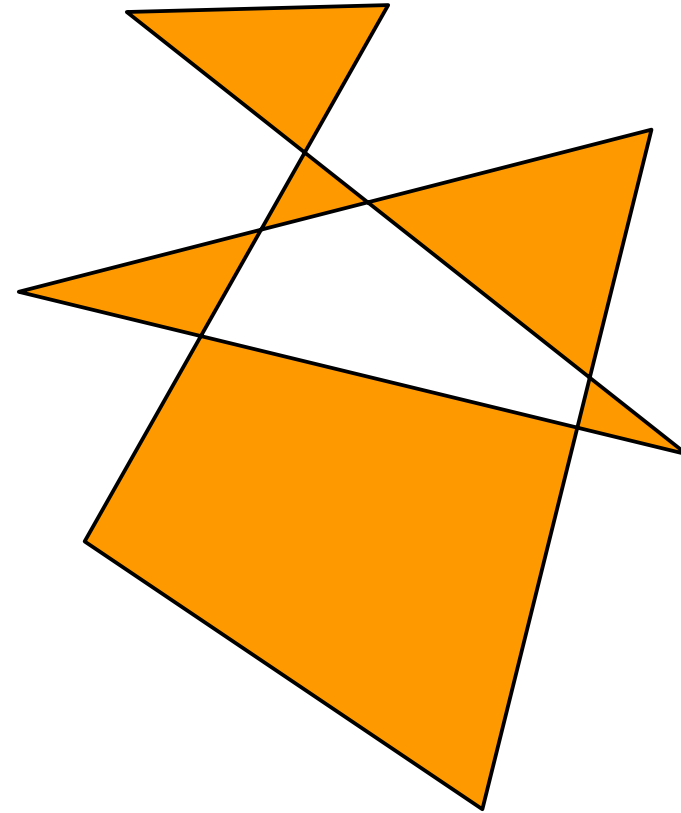
pepca@cgg.mff.cuni.cz
<https://cgg.mff.cuni.cz/~pepca/>



Pravidla vyplňování

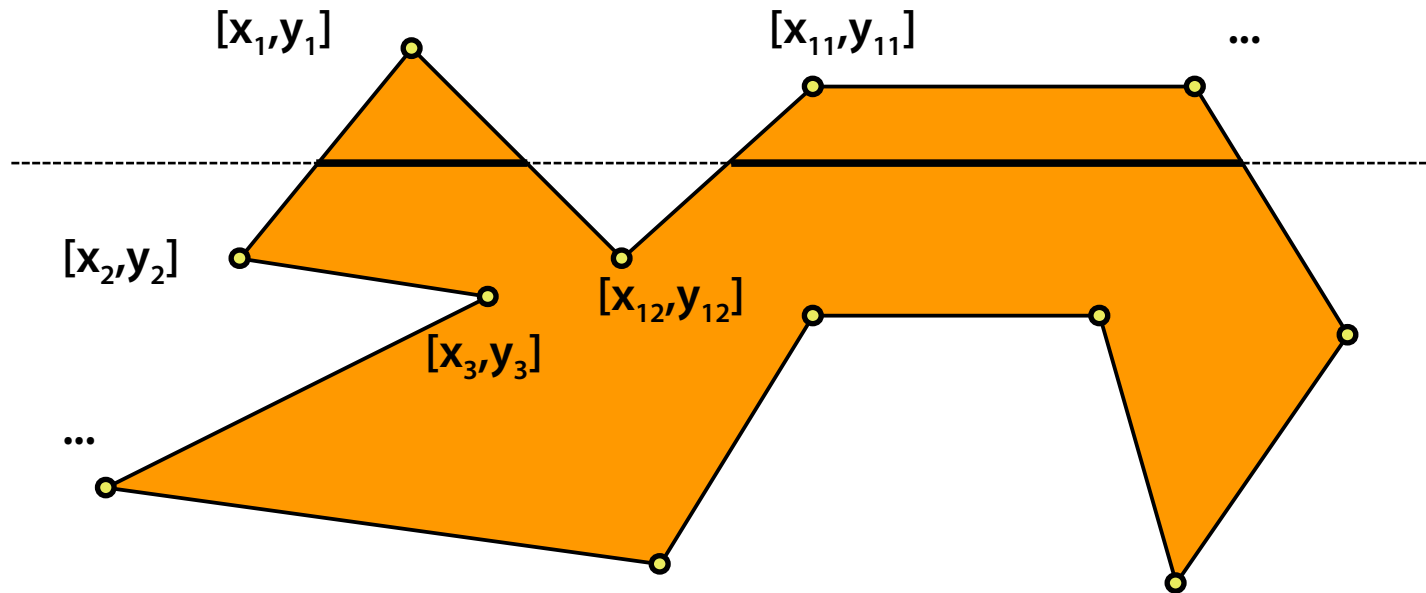


všechny vnitřní body



jen liché body
(„odd-even“ rule)

Řádkový algoritmus



N-úhelník je zadán posloupností svých vrcholů

Může být **nekonvexní**

Možné zjednodušení – vyplňují se jen **liché body**



1. předzpracování

N-úhelník rozložíme na **jednotlivé hrany**

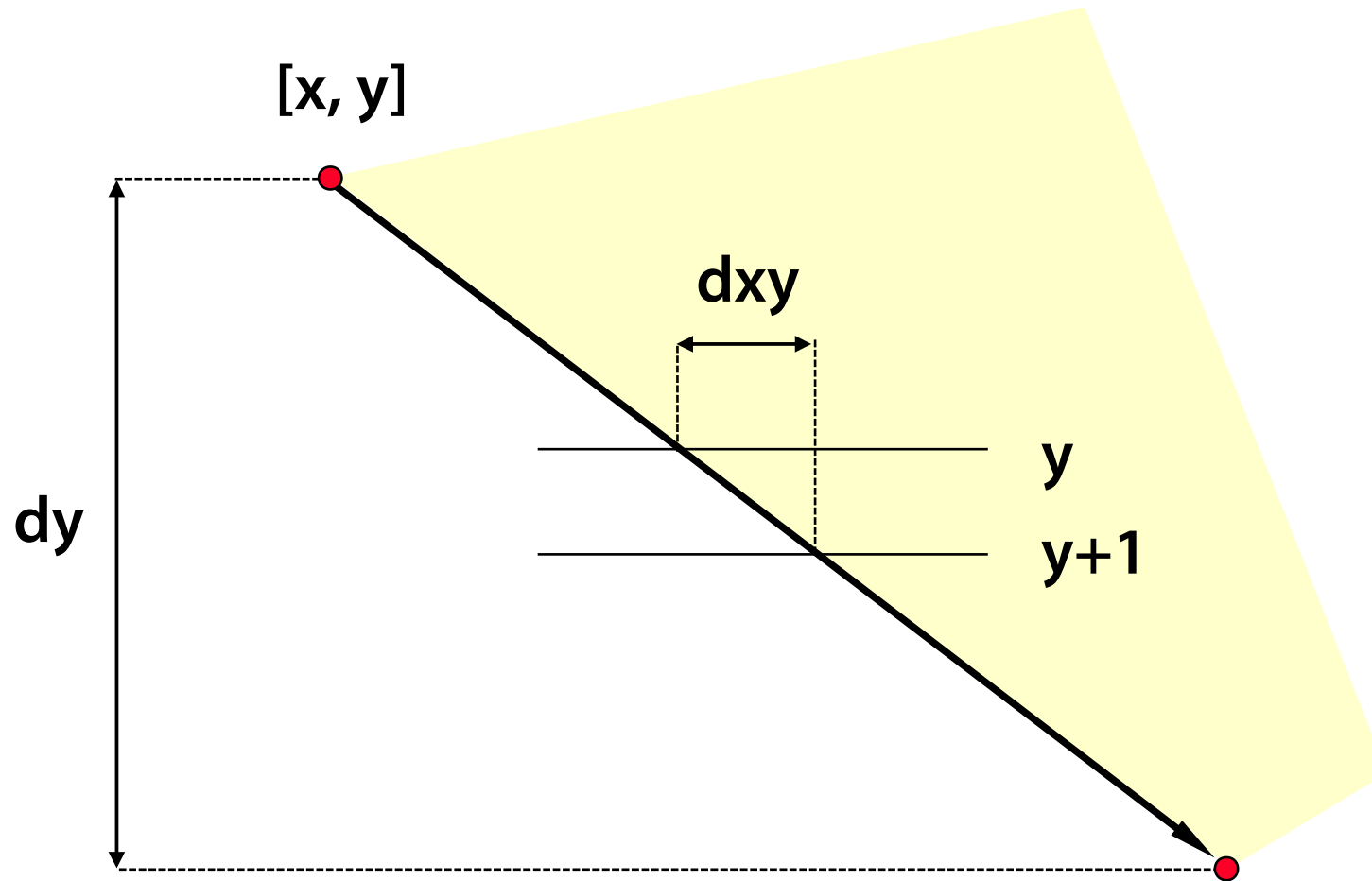
Vodorovné hrany odstraníme

Pro ostatní hrany vytvoříme **pracovní záznamy**

– hrany orientujeme směrem shora dolů



Pracovní záznam pro hranu





Pracovní záznam pro hranu

double x;

// souřadnice x horního koncového bodu,
// později souřadnice průsečíku s aktuální řádkou

int y;

// souřadnice y horního koncového bodu

int dy;

// výška hrany v pixelech: $|y_2 - y|$

double dxy;

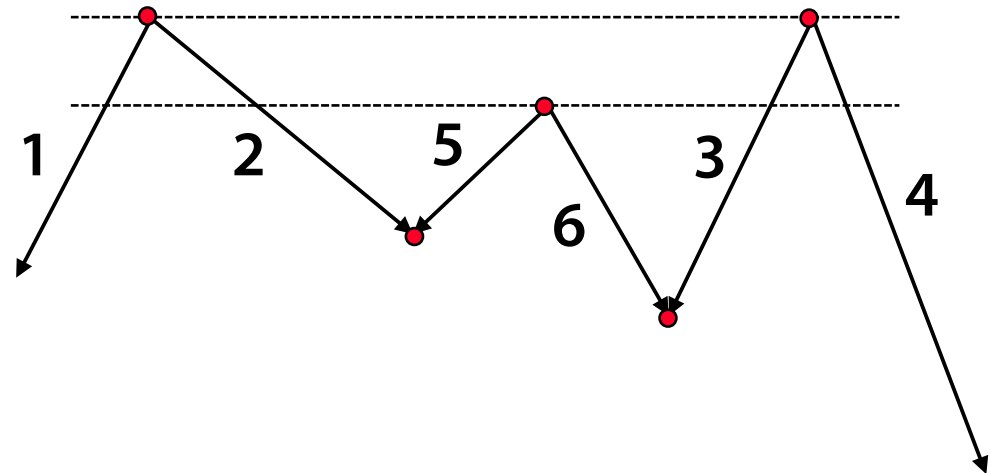
// změna x při posunutí na následující řádku (směrnice): $(x_2 - x) / dy$



2. inicializace seznamu S

Všechny předzpracované hrany setřídíme do **vstupního seznamu S** podle kritérií:

- ① vzestupně podle y
- ② vzestupně podle x
- ③ vzestupně podle d_{xy}

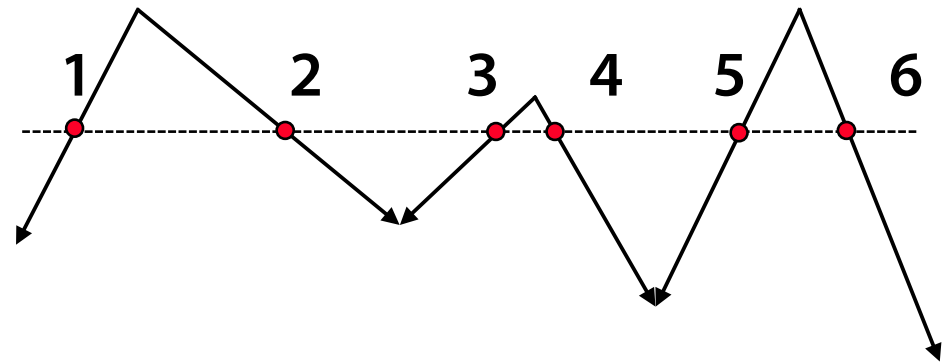




3. aktuální seznam A

Aktuální seznam A bude obsahovat všechny hrany, které protínají aktuální řádku. Seznam budeme udržovat setříděný:

- ② vzestupně podle x
- ③ vzestupně podle dxy



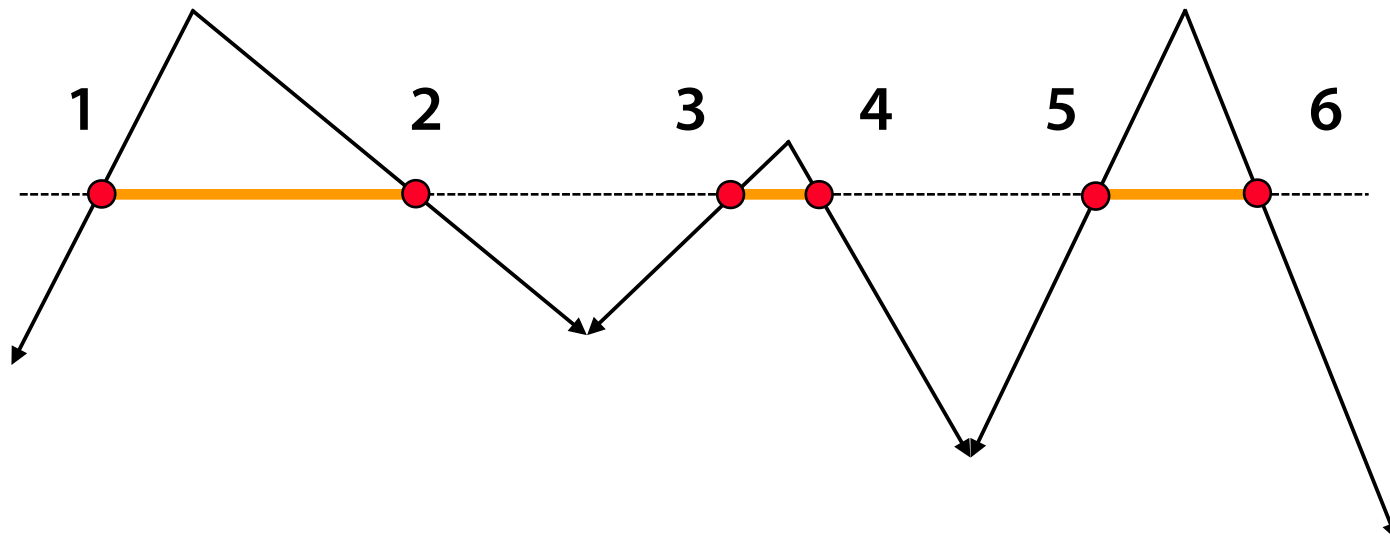
Při inicializaci zařadíme do A počáteční úsek seznamu S – hrany se shodným (tj. minimálním) y



4. vykreslení aktuální řádky

Je třeba projít **aktuální seznam A** a vykreslit úseky odpovídající vnitřku n-úhelníka

- kreslím každý úsek mezi lichým a sudým záznamem
- při jiném pravidle vyplňování by byly podmínky složitější...





5. přechod na další řádku

Aktualizace seznamu A

```
dy--;  
if (dy == 0)  
{ /* odstraň hranu ze seznamu A ... */  
}  
else  
    x += dxy;
```

Kontrola setřídění A

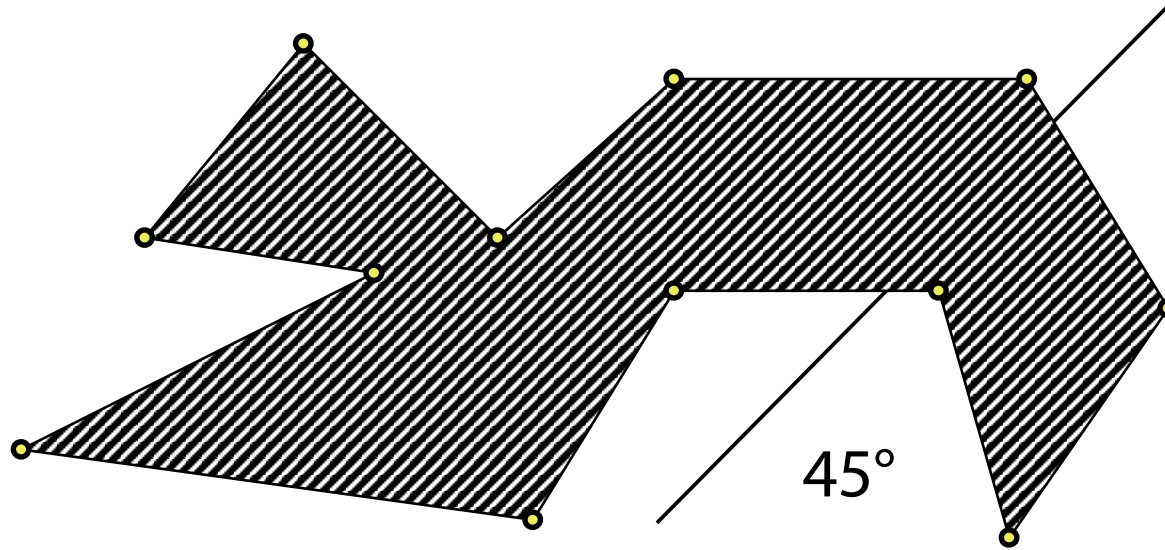
Zatřídění **nových hran** ze seznamu S do seznamu A
(počáteční úsek S)



6. podmínka ukončení cyklu

Jestliže je seznam A neprázdný, výpočet pokračuje
krokem 4

Jinak algoritmus **končí**

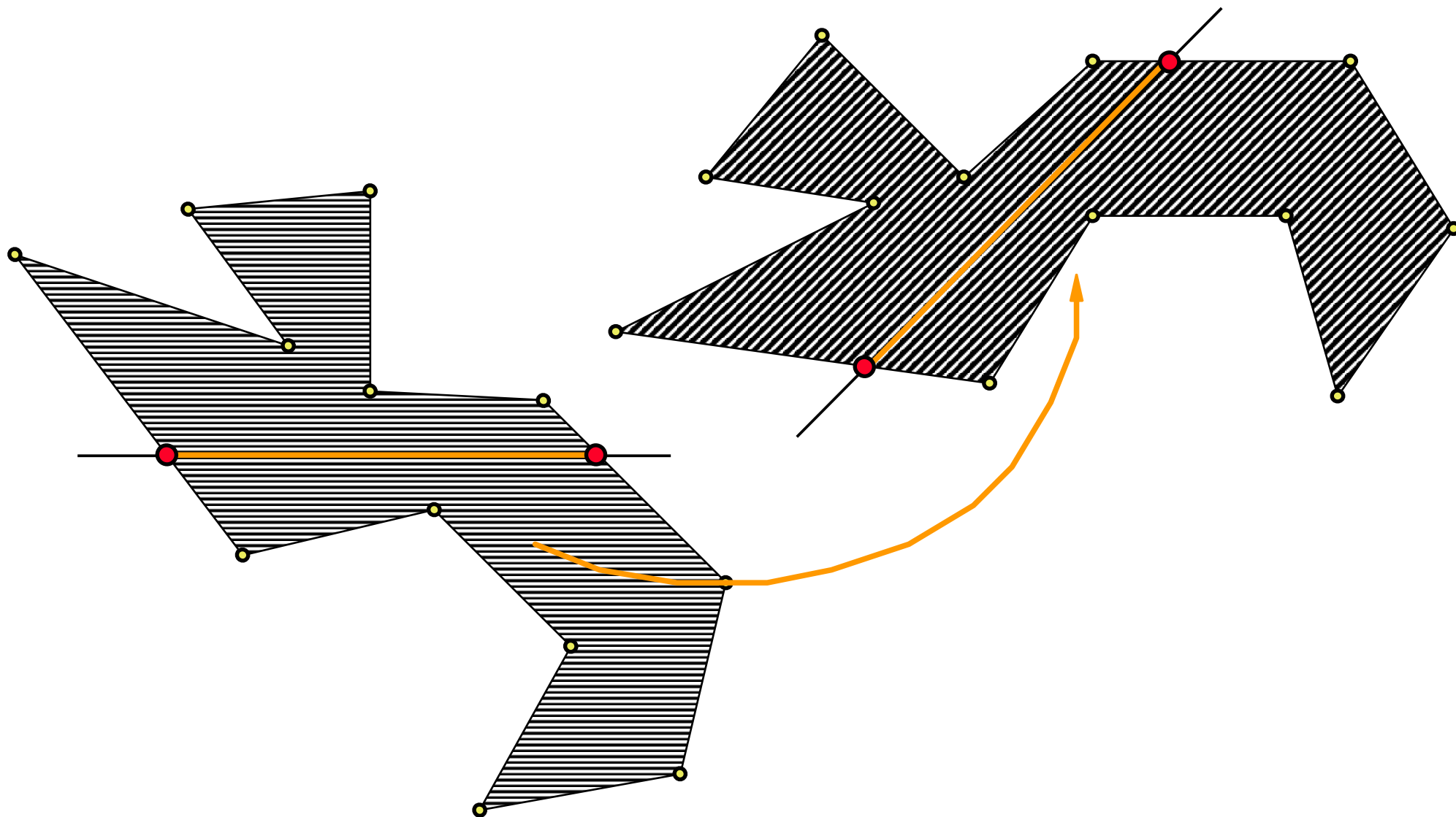


Otočím vrcholy n -úhelníka o opačný úhel

Kreslím každý **k-tý** řádek

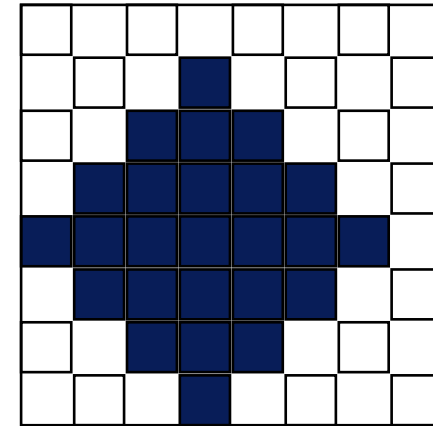
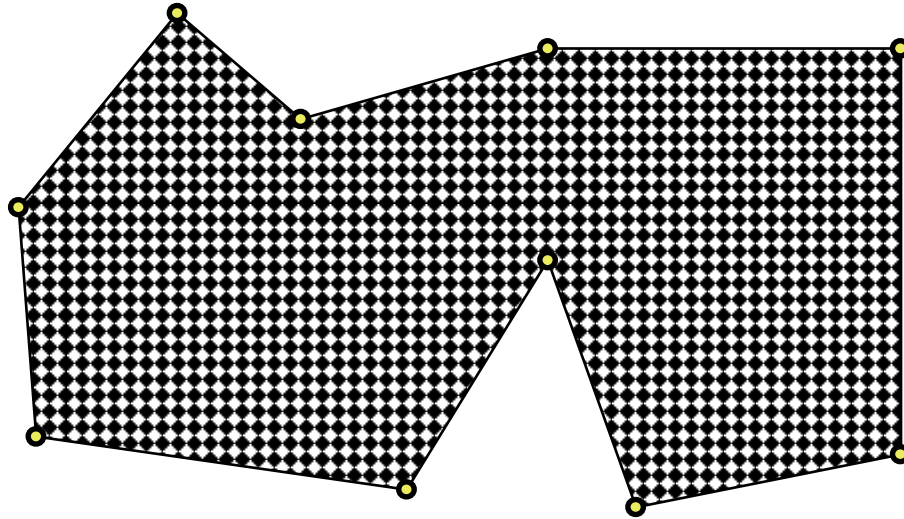
Před kreslením každou úsečku **otočím zpět**

Šrafovaní





Vyplňování vzorkem



matice $M[8,8]$

Vzorek je zadán **maticí pixelů** (např. 8×8)

Každý pixel se kreslí předpisem

– `PutPixel(x, y, M[y % 8, x % 8]);`



Literatura

J. Foley, A. van Dam, S. Feiner, J. Hughes: *Computer Graphics, Principles and Practice*, 92-99

**Jiří Žára a kol.: *Počítačová grafika, principy a algoritmy*,
129-138**