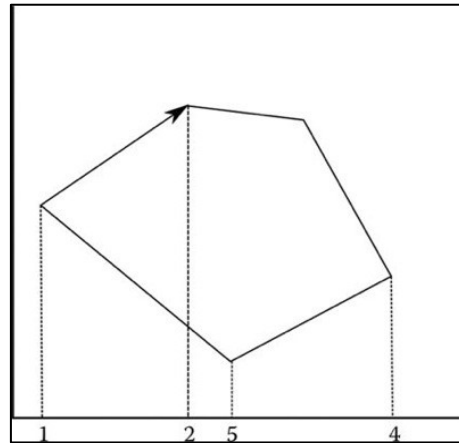
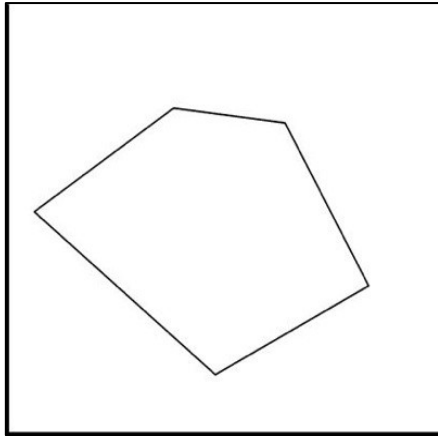


ÚLOHY S POLYGONEM

Polygon – řetězec úseček, poslední bod je totožný s prvním



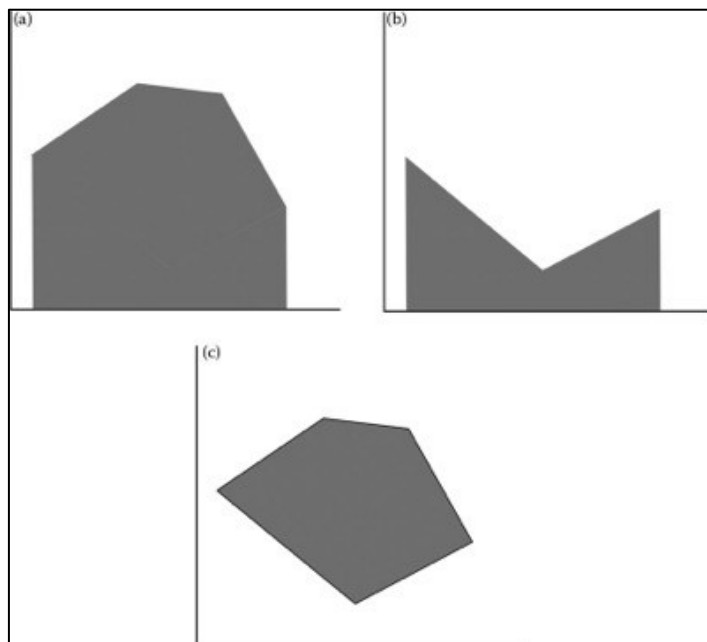
6 bodů: X_1, Y_1 až X_6, Y_6 $Y_1=X_6, Y_1=Y_6$

STANOVENÍ PLOCHY JEDNOHO POLYGONU

3 úsečky (segmenty) v horní části

2 úsečky ve spodní části

(plocha pod horními 3 úsečkami) – (plocha pod spodními 2 úsečkami) = plocha polygonu



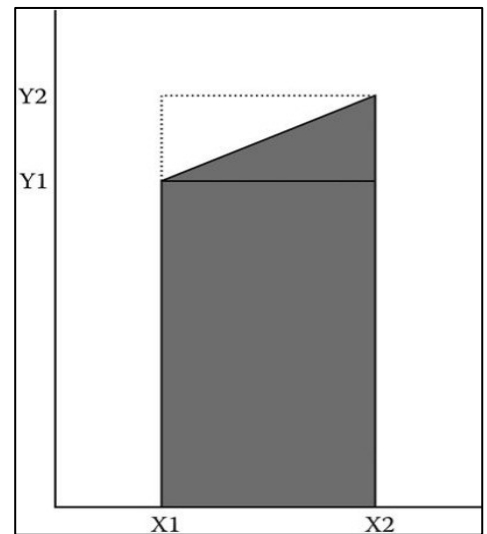
Bod	X	Y
1	1	8
2	2	10
3	4	9
4	5	5
5	3	2
6	1	8

Algoritmus pro výpočet plochy pod úsečkou:

1. plocha obdélníka: šířka $(X_2 - X_1)$ a výška $Y_1 \rightarrow \text{Plocha} = (X_2 - X_1) \cdot Y_1$
2. plocha trojúhelníka: $(X_2 - X_1) \cdot (Y_2 - Y_1) / 2$

Plocha pod úsečkou celkem: $(X_2 - X_1) \cdot (Y_2 + Y_1) / 2$

Pod úsečkou	plocha
1	18
2	38
3	14
4	-14
5	-20
Celkem polygon	36



Plochy pod spodními úsečkami mají zápornou hodnotu \rightarrow součet všech ploch pod úsečkami = plocha polygonu

Pokud řazení bodů bude proti směru hodinových ručiček, vyjde plocha s opačným znaménkem

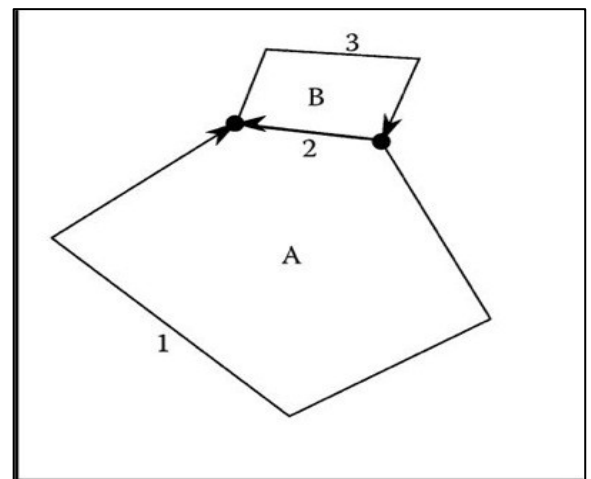
URČOVÁNÍ PLOCH VÍCE POLYGONŮ

při opakování se hraniční čáry budou brát dvakrát

Využití topologie: topologické tabulky
uzly, čáry, polygony

polygon B: čáry L2, L3
ve shodné orientaci (po směru hodin)
lze použít předchozí algoritmus

polygon A: čáry L1, L2
odlišná orientace úseček (vektorů)
orientaci lze poznat z topologických tabulek:
polygon vlevo, vpravo od čáry



pravidlo sečítání ploch pod úsečkami do celkové plochy polygonu:
stejná orientace s ostatními čarami - stejné znaménko
opačná orientace - změnit znaménko

koncept lze použít i na určení plochy „vnějšího polygonu“

URČENÍ POLOHY BODU UVNITŘ JEDNOHO POLYGONU

vztahy bodových a plošných dat

koncept možného algoritmu:

polopřímka z bodu X,Y protíná hranici polygonu v lichém počtu průsečíků → bod leží uvnitř

algoritmus:

```
n = 0
pro každou úsečku
    otestuj protnutí s polopřímkou
    je-li průsečík, pak n = n + 1
je-li n je liché, bod je uvnitř polygonu
```

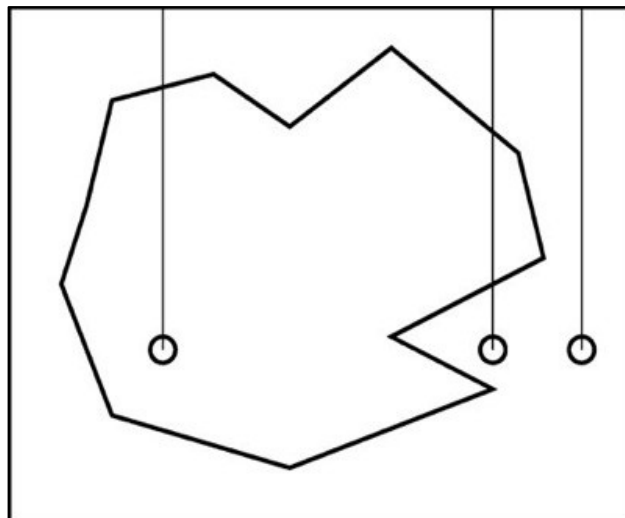
Vylepšený algoritmus:

```
n = -1
pro každou úsečku
    otestuj protnutí s polopřímkou
    je-li průsečík, pak n = -n
je-li n = +1 bod je uvnitř polygonu
```

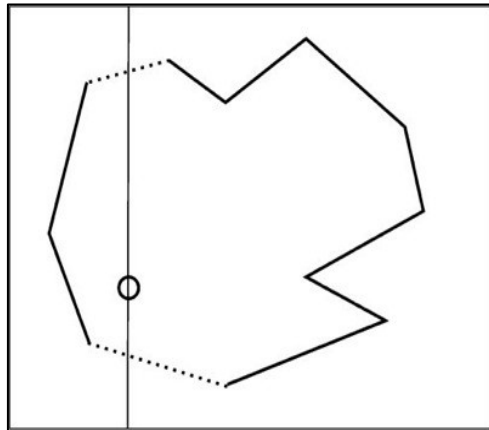
Zjišťování průsečíku

standardní metoda hledání průsečíků přímek – zjednodušeno vhodným zvolením testovací polopřímky z bodu X,Y

např. vertikální – > platí $x_p = X$, zbývá určit y_p

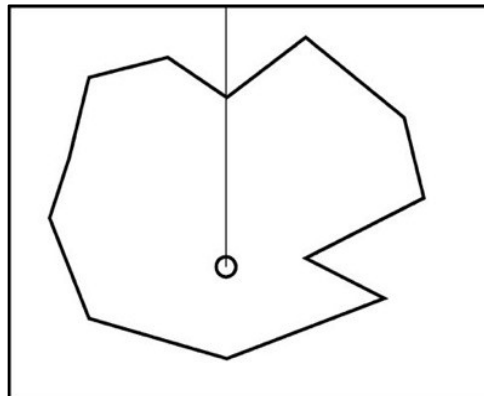


průsečík existuje jen když konce testované úsečky leží na různých stranách od přímky



$((X1 < xp) \text{ and } (X2 > xp)) \text{ or } ((X1 > xp) \text{ and } (X2 < xp))$

Pokud je průsečík shodný s počátečním bodem úsečky



přidána ještě jedna možnost:

$X1 = xp$ or $((X1 < xp) \text{ and } (X2 > xp)) \text{ or } ((X1 > xp) \text{ and } (X2 < xp))$

problém s přesností výpočtu při stanovení rovnosti (průsečík v krajním bodě úsečky)
proto místo rovnosti:

$(X1 > (xp - d) \text{ and } X1 < (xp + d))$

pro vyhovující úsečky standardním algoritmem stanovit souřadnici y_p a určit počet těch, pro které je $y_p > Y$ (zajímá nás počet průsečíků nad bodem – na polopřímce)

URČENÍ PŘISLUŠNOSTI BODU PRO VÍC POLYGONŮ

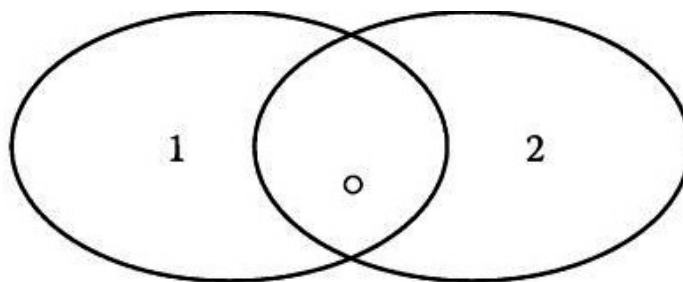
opakování postupu pro jeden polygon

OON polygonu - pokud bod uvnitř obdélníku polygon postupuje do testu zda bod i uvnitř polygonu, jinak vyřazen
snížení počtu možných polygonů až o dva řády

zbylé polygony jsou sousedící - většina čar (a jejich segmenty) bude brána dvakrát do výpočtu

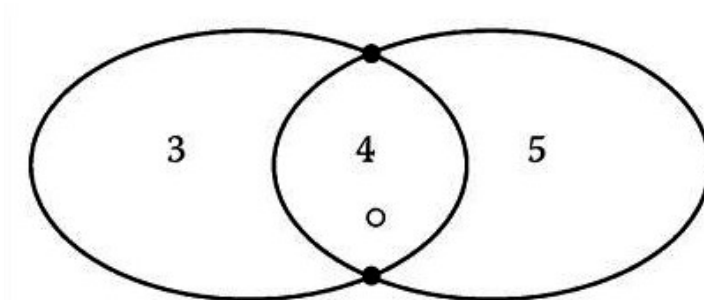
Využití topologie pro efektivnější algoritmus:

Bez topologie průsečík není v databázi zanesen jako uzel:
stejná čára může vyjadřovat hranici mezi polygonem 2 a vnitřkem polygonu 1 ale také mezi polygonem 2 a jeho vnějškem.



Bez topologie bod může ležet uvnitř několika polygonů a nebo v žádném.

S topologií průsečík je v databázi zanesen jako uzel:

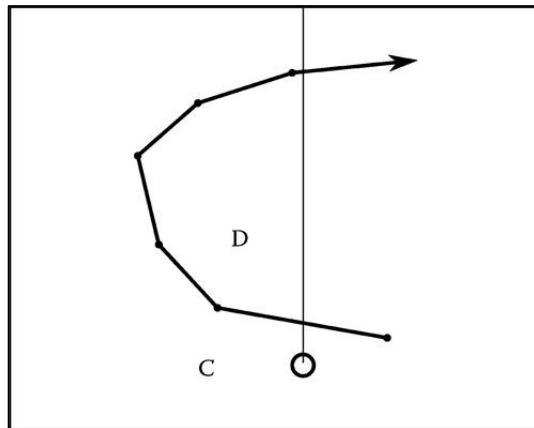


Topologická pravidla využívaná pro tuto operaci v GIS:

- pokud jsou čáry v rovině a kříží se, musí se protínat
- průsečík čar musí být uzel - vzniknou nové polygony v databázi
- každá čára musí ležet mezi nějakými dvěma polygony - "vnější" polygon

S topologií bod leží vždy uvnitř nějakého polygonu (jen v jednom polygonu pokud neuvažujeme bod na hranici a vnořené polygony)

vertikální polopřímka z bodu nahoru protne jako první čáru, která je hranicí mezi dvěma polygony
--> uvnitř jednoho leží bod



Dvě fáze testu:

1. nalézt první úsečku, kterou polopřímka protne
2. rozhodnout ve kterém ze dvou polygonů bod leží

Řešení pro 1: Polopřímka je vertikální --> první průsečík má nejmenší souřadnici Y namísto všech vytvoření NOO pro každou úsečku a hledat průsečík jen pro vybrané úsečky.

Řešení pro 2: Polygon obsahující bod je ten, který leží "pod" úsečkou s prvním průsečíkem.

Rozhodnutí z topologie, podle směrové orientace protnuté úsečky: při směru P-->L je polygon vlevo, při směru L-->P je polygon vpravo

Obecné poznatky k tvorbě algoritmu:

- Algoritmy pro zdánlivě jednoduché úlohy mohou být dost složité
- Datová struktura hraje důležitou roli pro tvorbu algoritmu
- Algoritmy se liší dobou potřebnou na jejich provedení