

Malířův algoritmus

© 1995-2019 Josef Pelikán
CGG MFF UK Praha

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



Malířův algoritmus

Kreslení do bufferu

- video-RAM, GPU, rastrová tiskárna s bufferem

Vyplňování ploch

- lze i stínovat

Kreslení odzadu dopředu

- překreslování dříve nakreslených objektů
- kreslení **poloprůhledných objektů na GPU**

→ Určení správného pořadí ploch



Zjednodušené varianty

Explicitní pořadí kreslení

- např. u grafu funkce dvou proměnných: $z = f(x,y)$

Hlubkové třídění („depth-sort“)

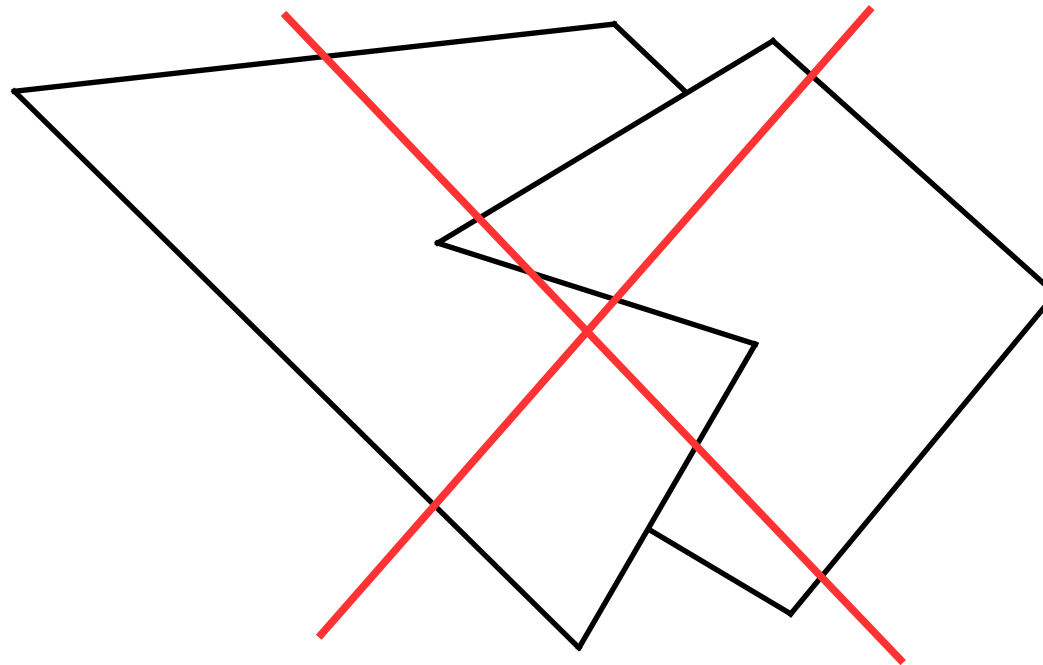
- **setřídění objektů podle souřadnice z** (střed, těžiště)
- dobře funguje při velkém množství malých objektů
- nesprávná kresba velkých ploch (velká stolní deska s malými předměty)



Korektní algoritmus

Scéna je složena z **rovinných plošek (stěn)**

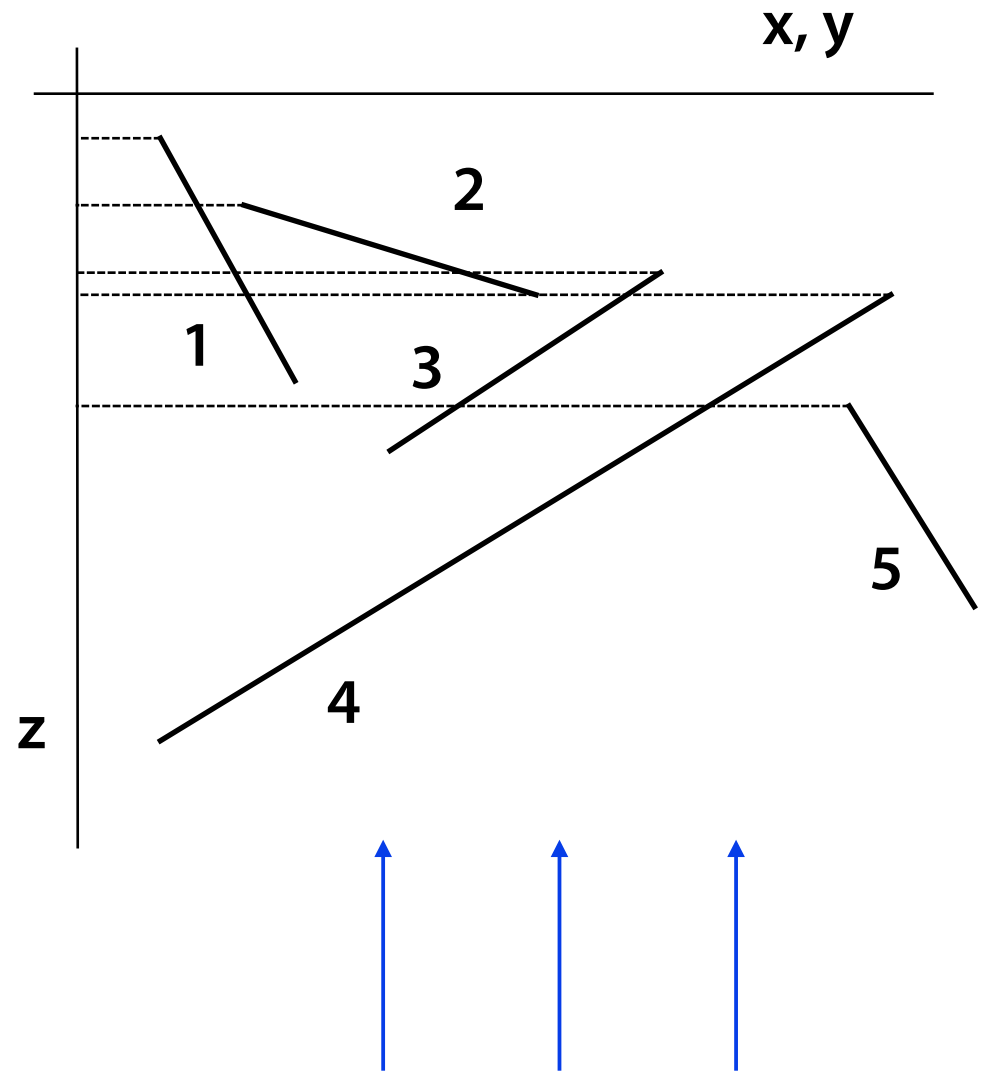
- stěny mohou mít společné body **pouze na obvodu** (**nesmějí se prosekávat**)





1. fáze: třídění

Stěny setřídíme podle **minimální souřadnice z** **vzestupně** – tj. odzadu dopředu – vytvoříme tak **vstupní seznam S**



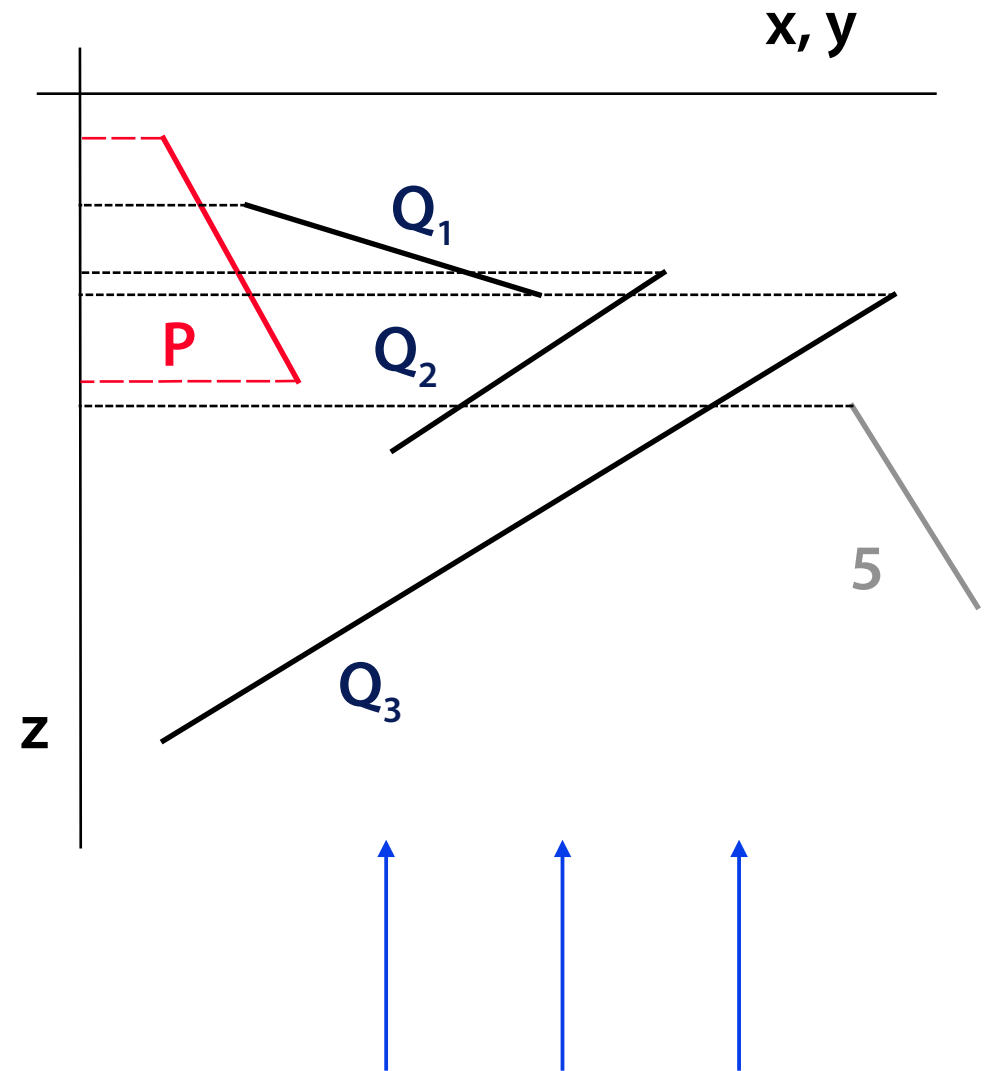


2. fáze: kontrola pořadí

Ze začátku seznamu S vezmeme stěnu P – **kandidáta** pro kresbu.

Proti P musíme otestovat ostatní stěny, které s ní mohou kolidovat.

- právě testovanou stěnu označíme $Q_{[i]}$

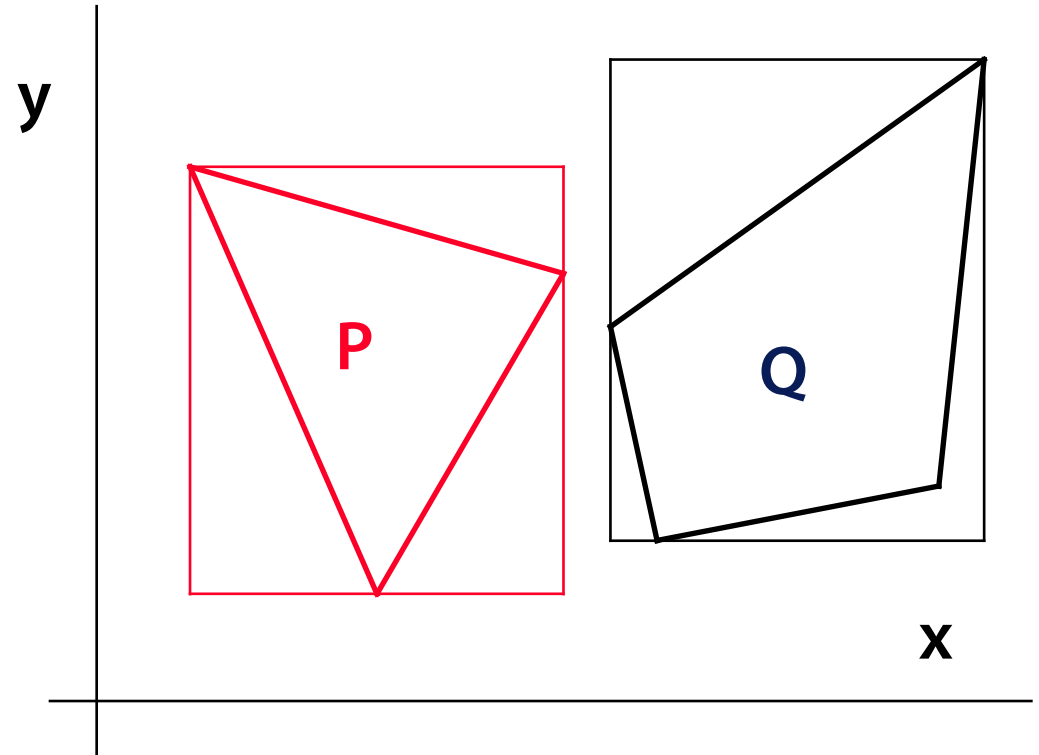




2.a fáze: „minimax test“

Nejprve provedeme
nejjednodušší test –
v **průmětu** porovnáme
obdélníky opsané oběma
stěnám

- jestliže nemají společný bod, testování **Q** končí (pass)
- jinak pokračujeme dalším testem **P** a **Q**

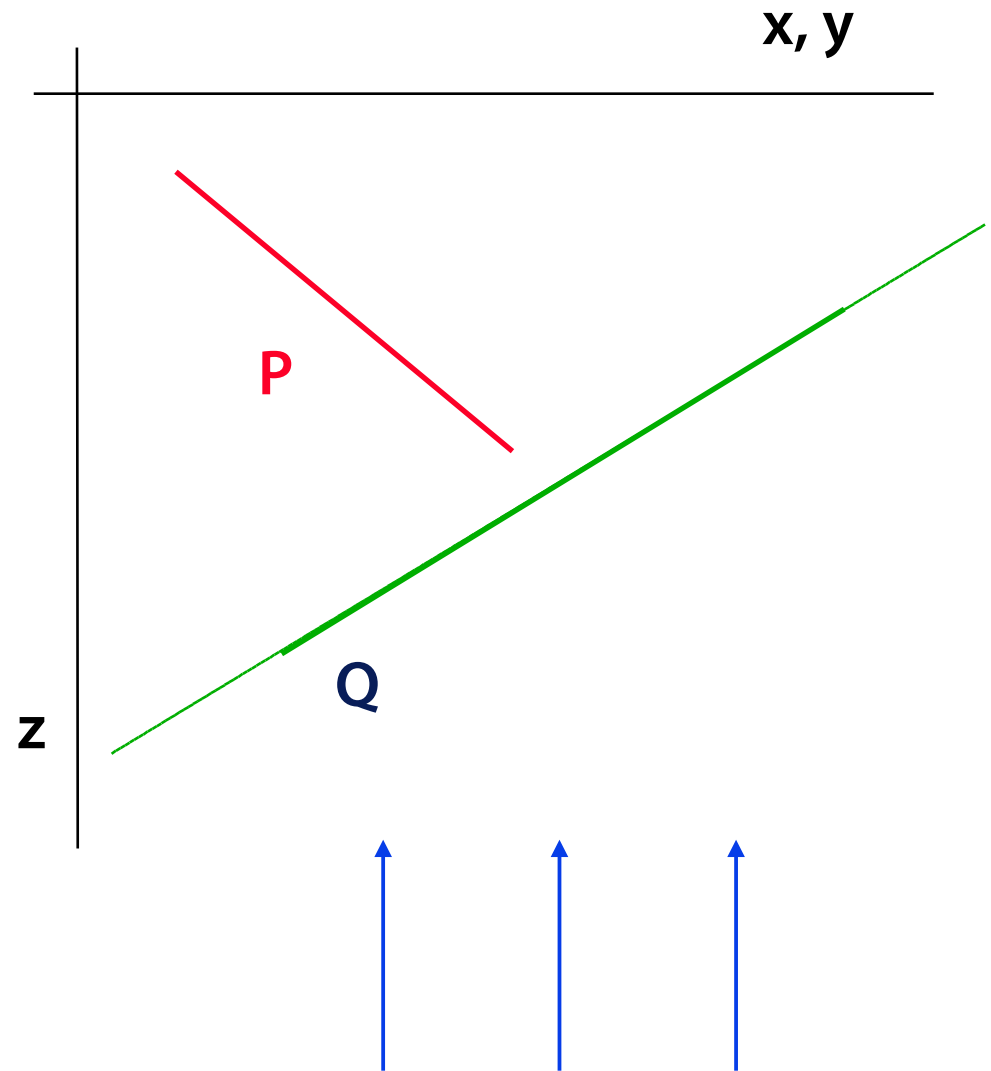




2.b fáze: P versus rovina Q

Testujeme, zda stěna **P**
neleží celá **za rovinou**
stěny **Q**

- v kladném případě
testování **Q** končí (pass)
- jinak pokračujeme dalším
testem **P** a **Q**



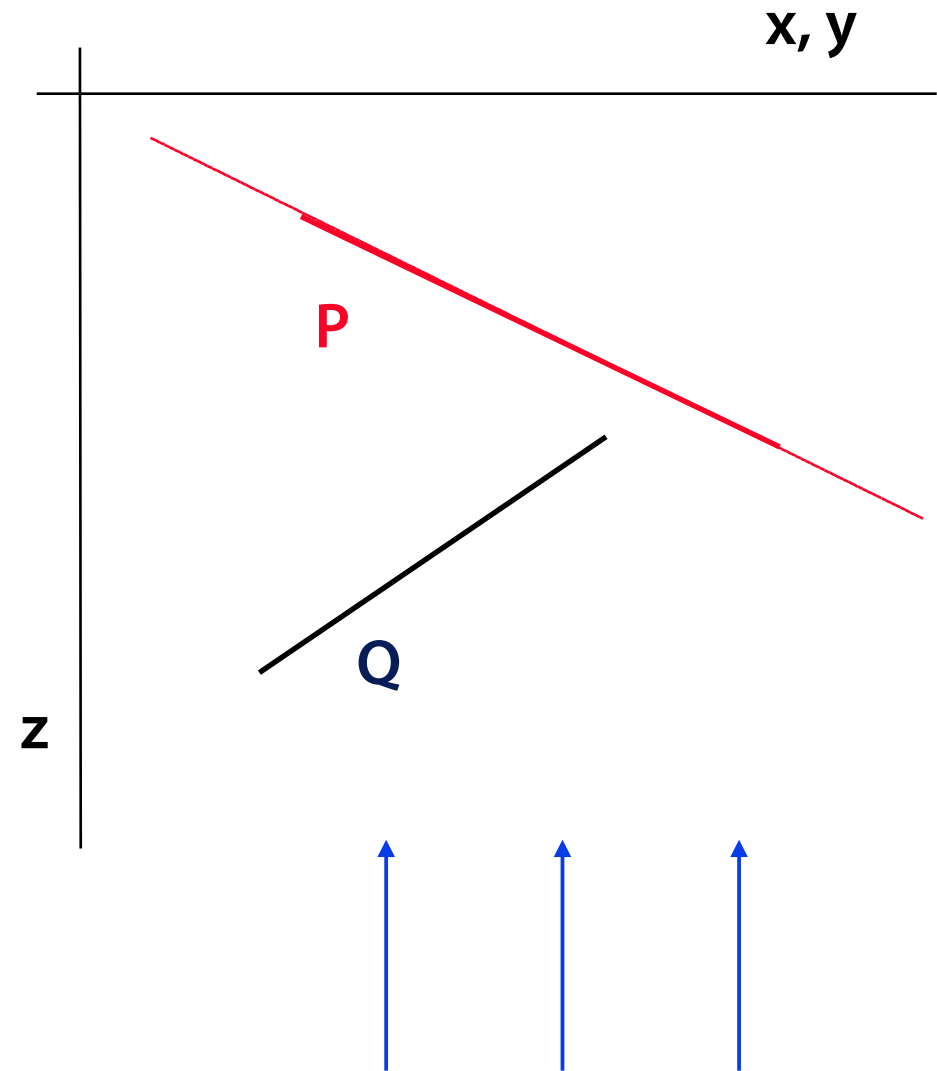
$$a \cdot x + b \cdot y + c \cdot z + d < 0$$



2.c fáze: Q versus rovina P

Test, zda stěna Q neleží celá před rovinou stěny P

- v kladném případě testování Q končí (pass)
- jinak pokračujeme dalším testem P a Q



$$a \cdot x + b \cdot y + c \cdot z + d > 0$$

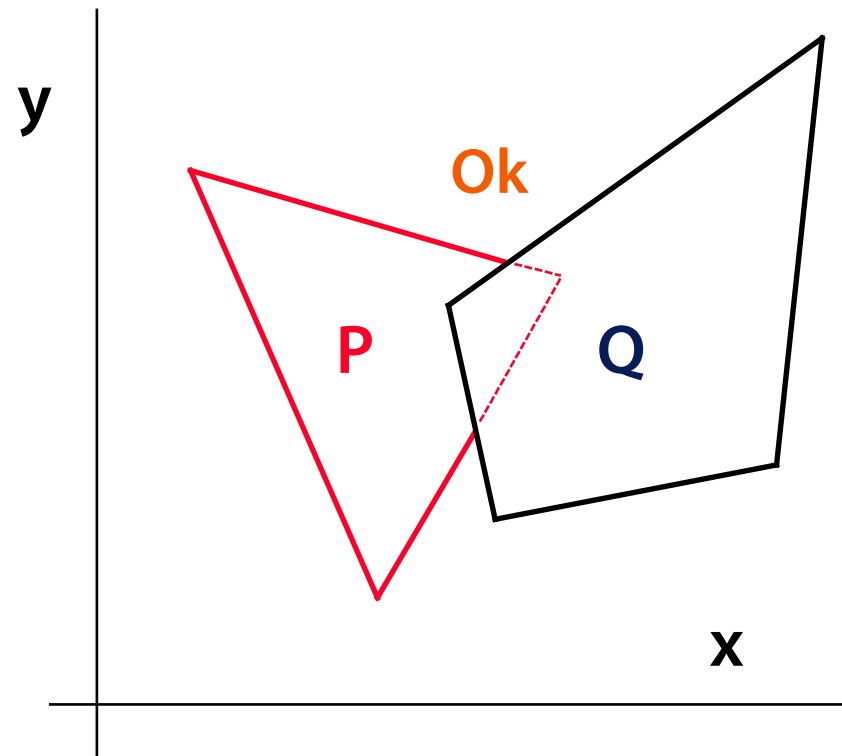


2.d fáze: úplný test v průmětu

Pokud předchozí testy neuspěly, musíme provést **úplný test stěn P a Q v průmětu**

Je potřeba zjistit, zda není některá část **Q** překrytá stěnou **P**

- v takovém případě by nešlo nakreslit **P** před **Q**!

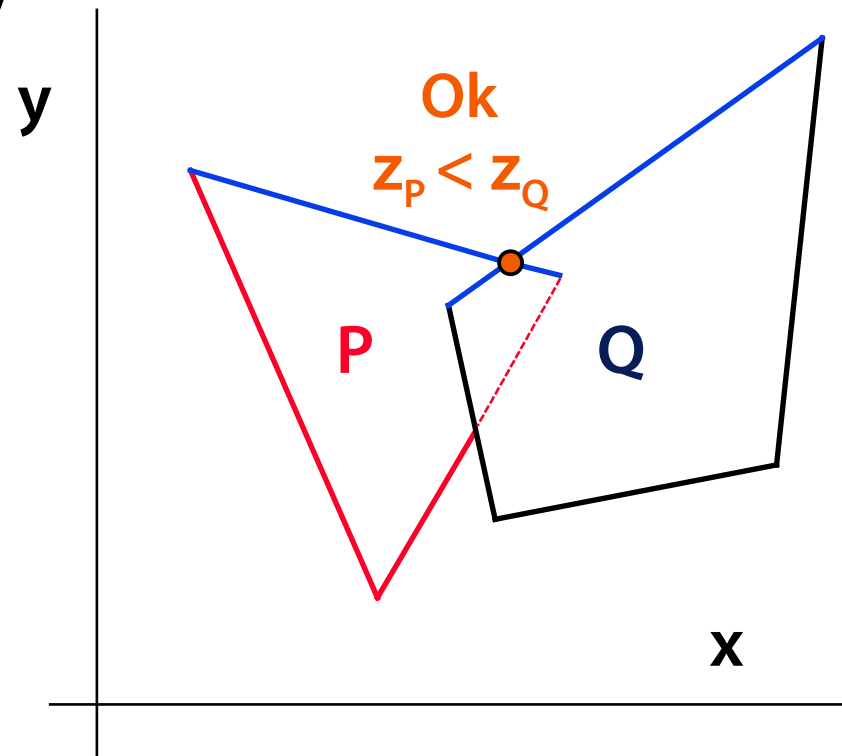




2.d fáze: úplný test v průmětu

Testujeme proti sobě **všechny** hrany P a Q

- najdeme-li průsečíky, porovnáme v nich souřadnice z
- je-li vždy P za Q, test Q končí úspěšně (pass)
- v opačném případě **nelze P nakreslit jako první!**

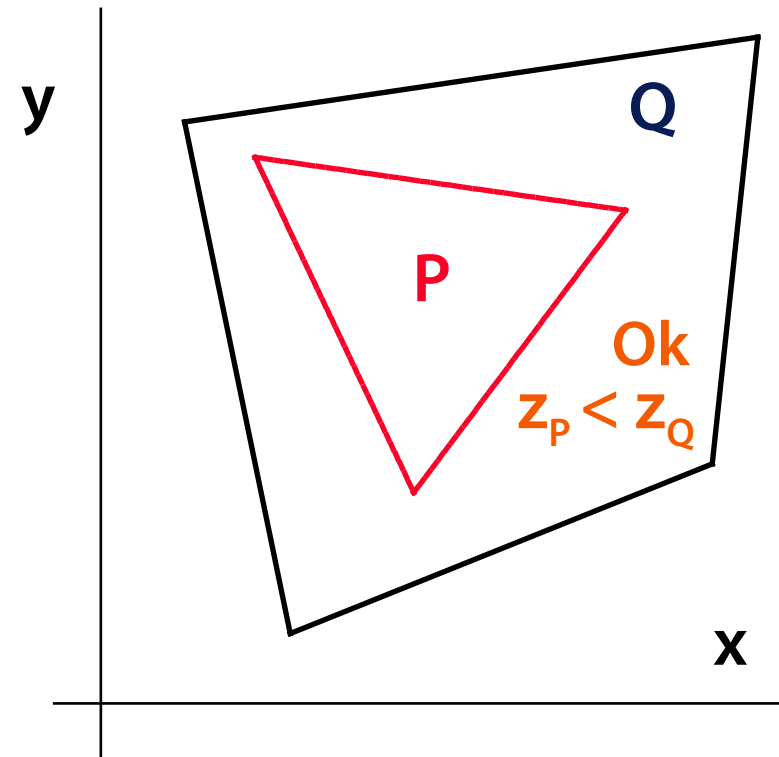




2.d fáze: úplný test v průmětu

Neexistuje-li průsečík hran **P** a **Q**, je třeba ještě zkontrolovat, zda neleží stěna **P** celá uvnitř **Q** nebo naopak

- to by se opět musely testovat souřadnice **z**





2. fáze: změna pořadí

Jestliže nelze z nějakého důvodu nakreslit P před Q , zkusíme přesunout stěnu Q na začátek seznamu S (ještě před P)

- pro Q budeme opět provádět všechny testy 2. fáze (jak jsme je popsali se stěnou P)
- testy nového kandidáta Q proti P už byly z velké části provedeny, stačí pouze doplnit obrácené testy **2.b** a **2.c**

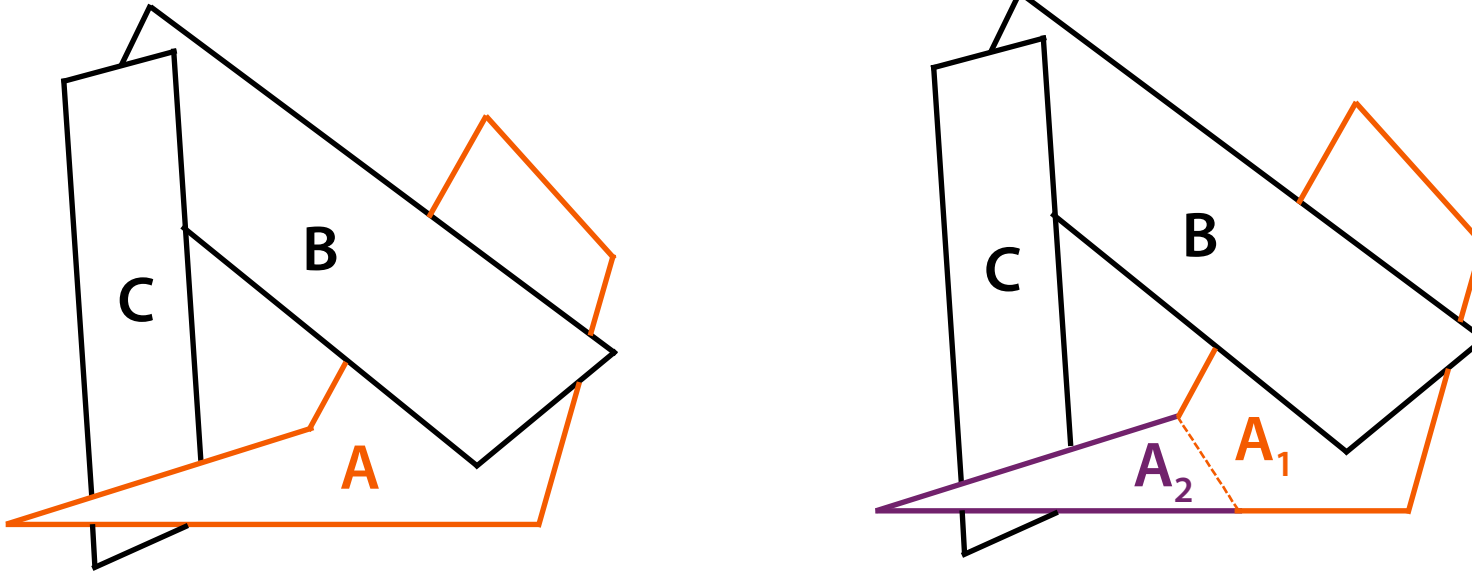
$P \rightarrow Q_1 \rightarrow Q_2 \rightarrow Q_3 \rightarrow 5$

$Q_1 \rightarrow P \rightarrow Q_2 \rightarrow Q_3 \rightarrow 5$

Kvůli možnosti **zacyklení** se musí každý kandidát označit zvláštním příznakem



2. fáze: zacyklení



Jestliže je testován některý kandidát podruhé, došlo k **zacyklení**

Cyklus lze odstranit **rozdělením** některé stěny

– správné pořadí: A_1, B, C, A_2



Literatura

J. Foley, A. van Dam, S. Feiner, J. Hughes: *Computer Graphics, Principles and Practice*, 672-675

Jiří Žára a kol.: *Počítačová grafika, principy a algoritmy*, 302-304