Malířův algoritmus

© 1995-2001 Josef Pelikán

KSVI MFF UK Praha

e-mail: Josef.Pelikan@mff.cuni.cz

WWW: http://cgg.ms.mff.cuni.cz/~pepca/

Malířův algoritmus

- **kreslení do bufferu**
  - video-RAM, rastrová tiskárna s bufferem
- **vyplňování ploch**
  - lze i stínovat

- **kreslení odzadu dopředu**
  - **překreslování** dříve nakreslených objektů

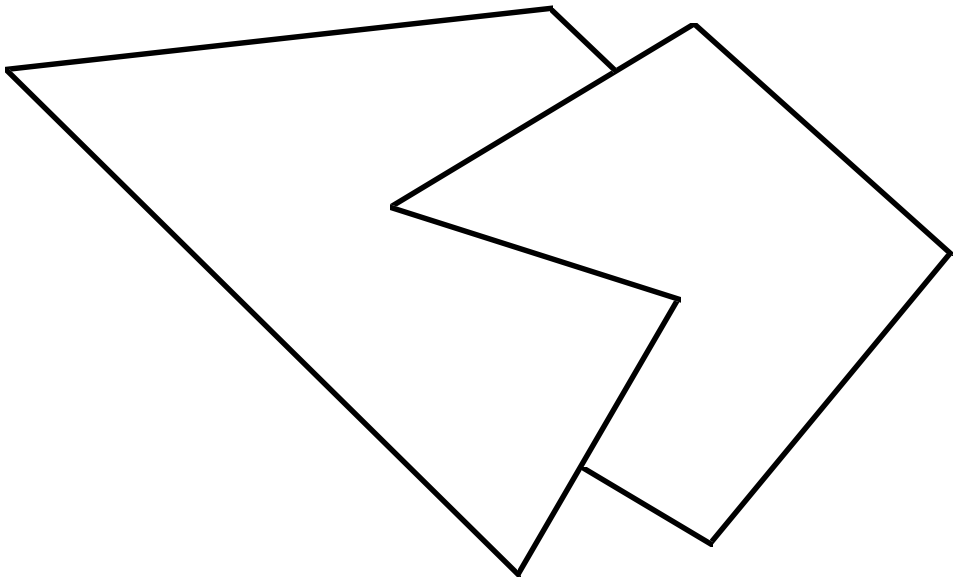
- **určení správného pořadí ploch**

Zjednodušené varianty

- **explicitní pořadí kreslení**
  - např. funkce dvou proměnných: $z = f(x,y)$
- **hloubkové třídění** (“depth-sort”)
  - **setřídění objektů** (plošek) **podle souřadnice *z*** (střed, těžiště)
  - dobře funguje při velkém množství malých objektů
  - nesprávná kresba velkých ploch (velká stolní deska s malými předměty)

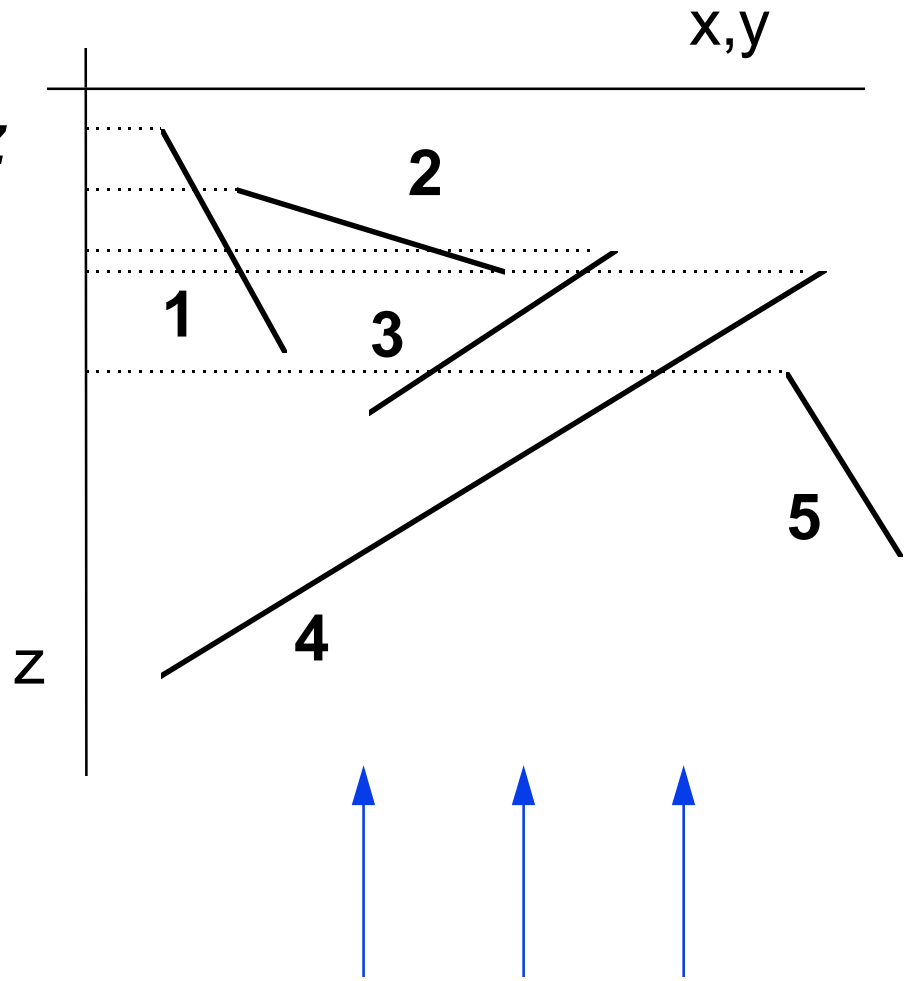
Korektní algoritmus:

- scéna je složena z **rovinných plošek**
- plošky mohou mít společné body **pouze na obvodu** (nesmějí se prosekávat)



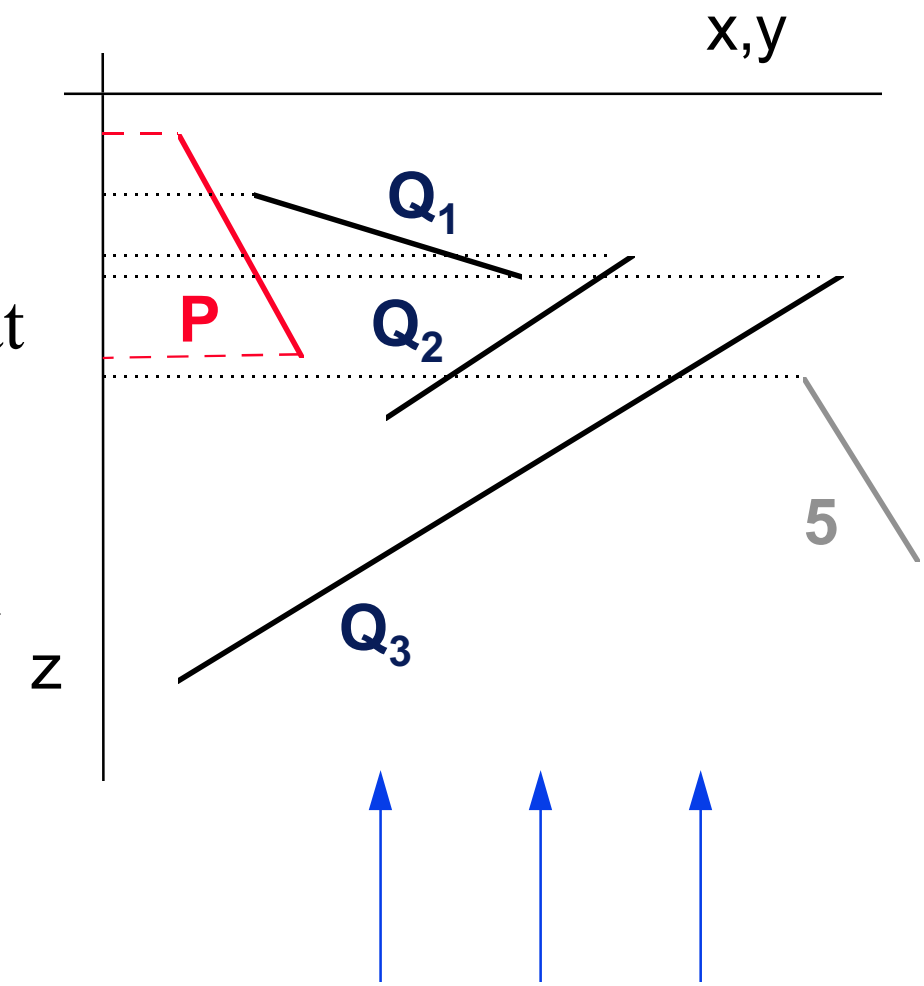
1. fáze: třídění

① plošky setřídíme podle **minimální souřadnice *z* vzestupně** - tj. odzadu dopředu - vytvoříme tak **vstupní seznam *S***



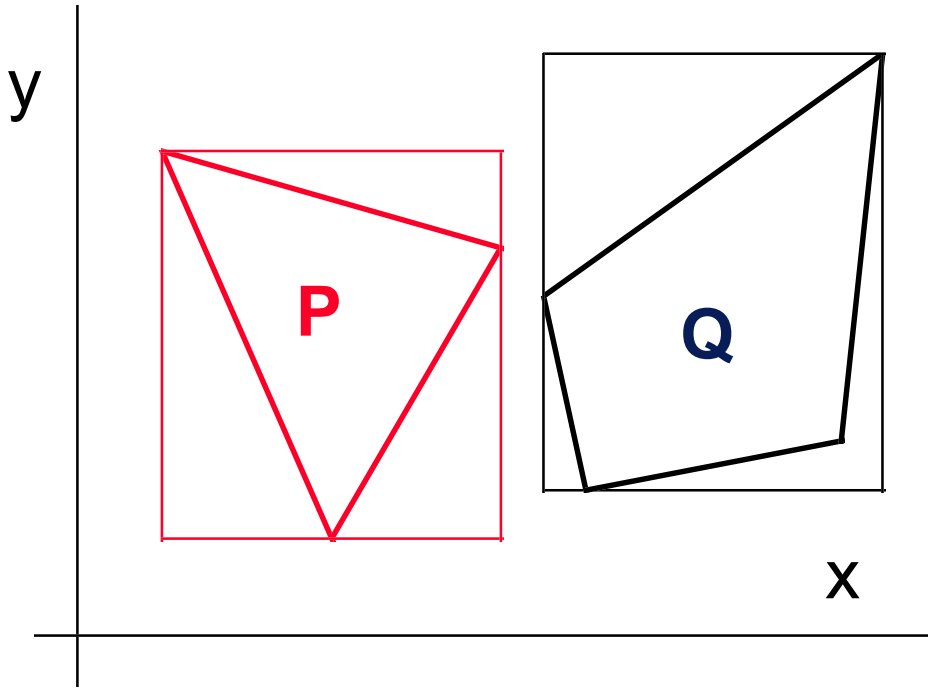
2. fáze: kontrola pořadí

② ze začátku seznamu ***S*** vezmeme plošku ***P*** - **kandidáta** pro kresbu. Proti ***P*** musíme otestovat ostatní plošky, které s ní mohou kolidovat. Právě testovanou plošku označíme ***Q***



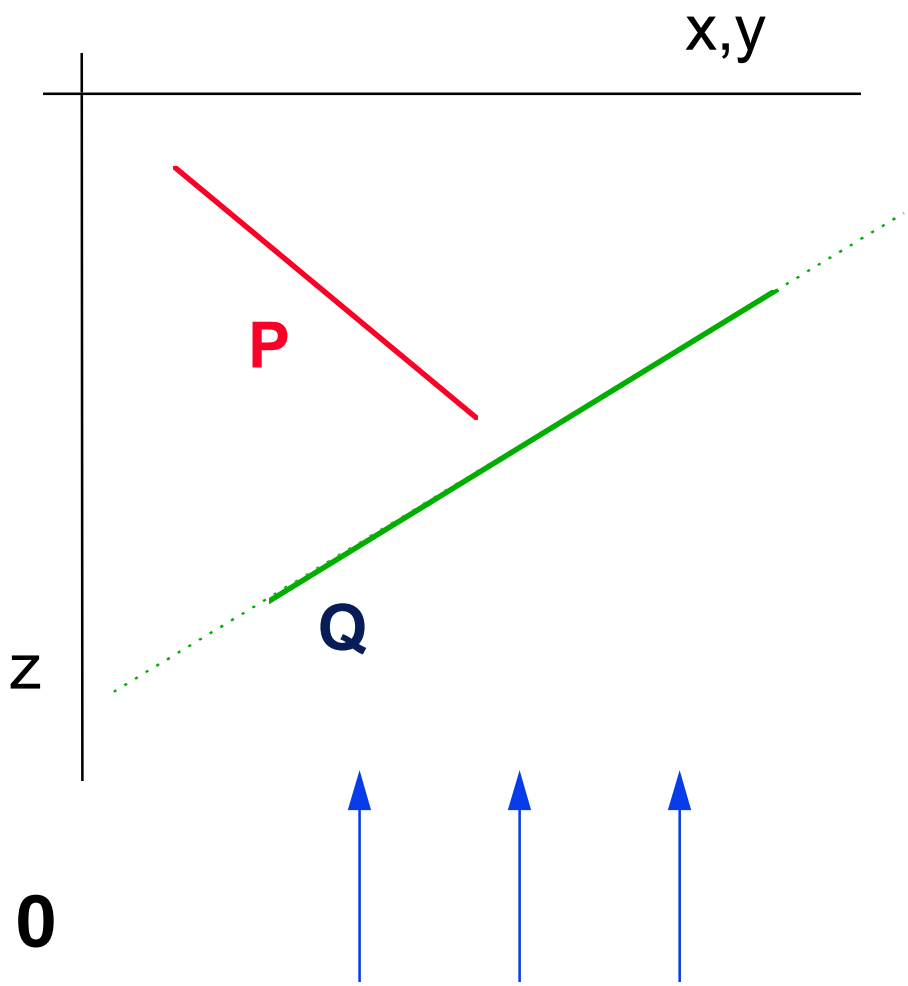
2.A fáze: “minimax test”

1. nejprve provedeme **nejjednodušší test - v průmětu** porovnáme obdélníky opsané oběma ploškám. Jestliže nemají společný bod, testování ***Q*** končí.
   Jinak pokračujeme dalším testem ***P*** a ***Q***



2.B fáze: *P* versus rovina *Q*

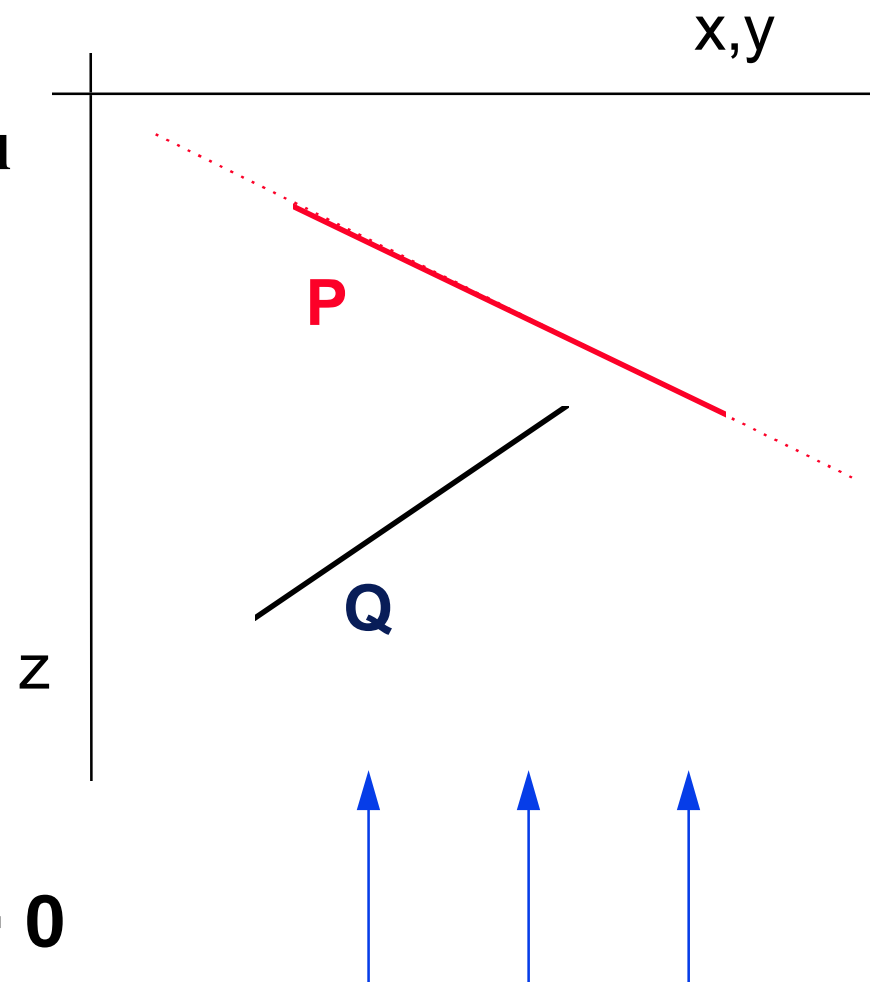
❷ dále testujeme, zda ploška ***P*** neleží celá **za rovinou** danou ploškou ***Q***. V kladném případě testování ***Q*** končí. Jinak pokračujeme dalším testem ***P*** a ***Q***



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

2.C fáze: ***Q*** versus rovina ***P***

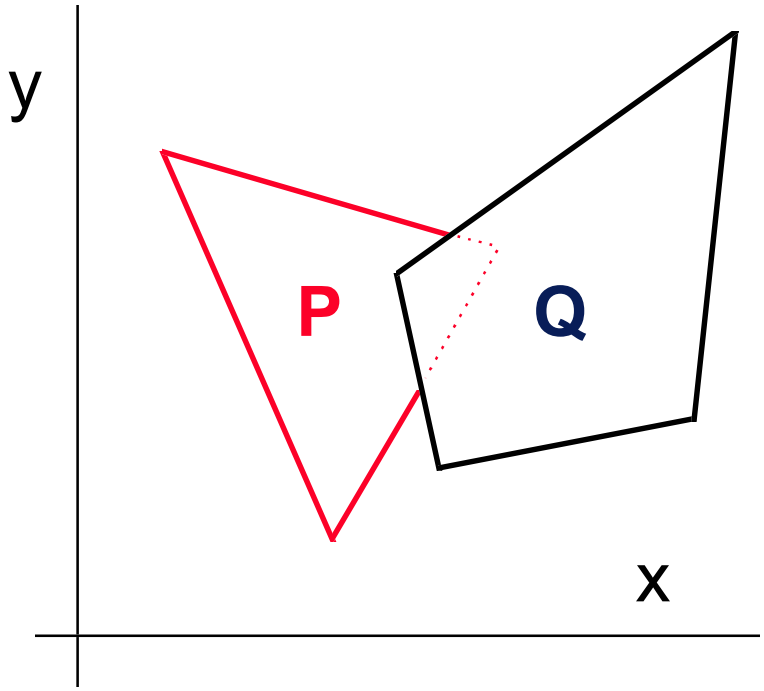
❸ testujeme, zda ploška ***Q*** neleží celá **před rovinou** danou ploškou ***P***.
V kladném případě testování ***Q*** končí.
Jinak pokračujeme dalším testem ***P*** a ***Q***



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

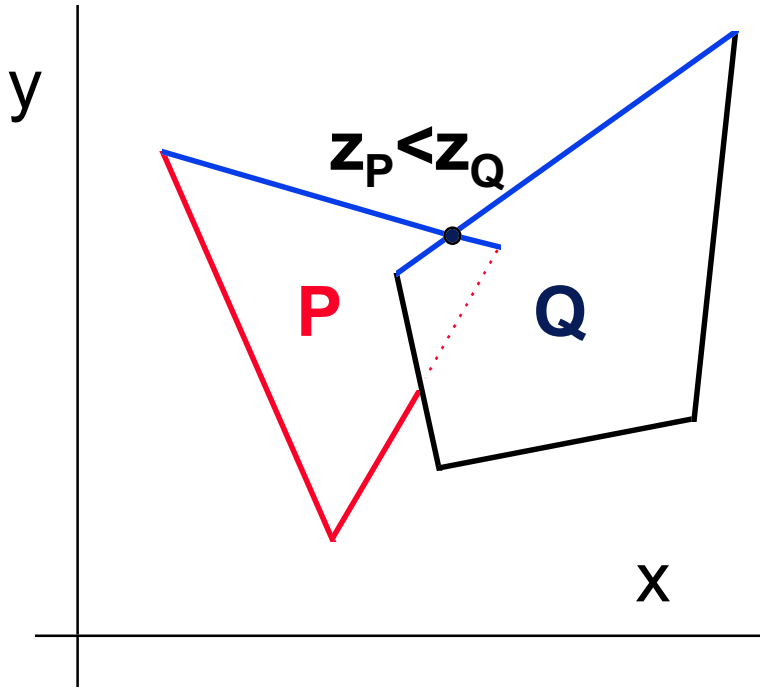
2.D fáze: úplný test v průmětu

❹ pokud předchozí testy neuspěly, musíme provést **úplný test** plošek ***P*** a ***Q*** **v průmětu**.
Je potřeba zjistit, zda není některá část ***Q*** překrytá ploškou ***P***.
V takovém případě by nešlo nakreslit ***P*** před ***Q***!



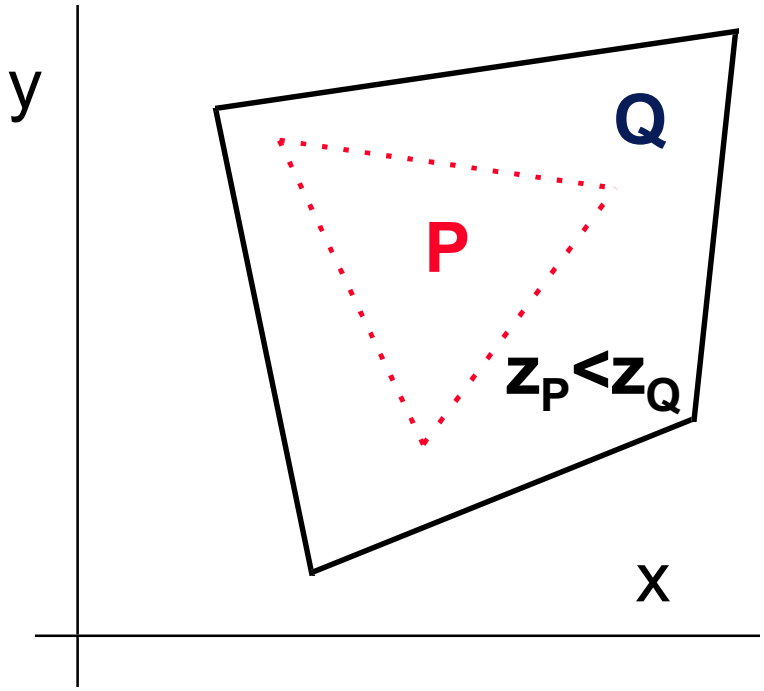
2.D fáze: úplný test v průmětu

- 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. Jinak nelze ***P*** nakreslit před ***Q***!



2.D fáze: úplný test v průmětu

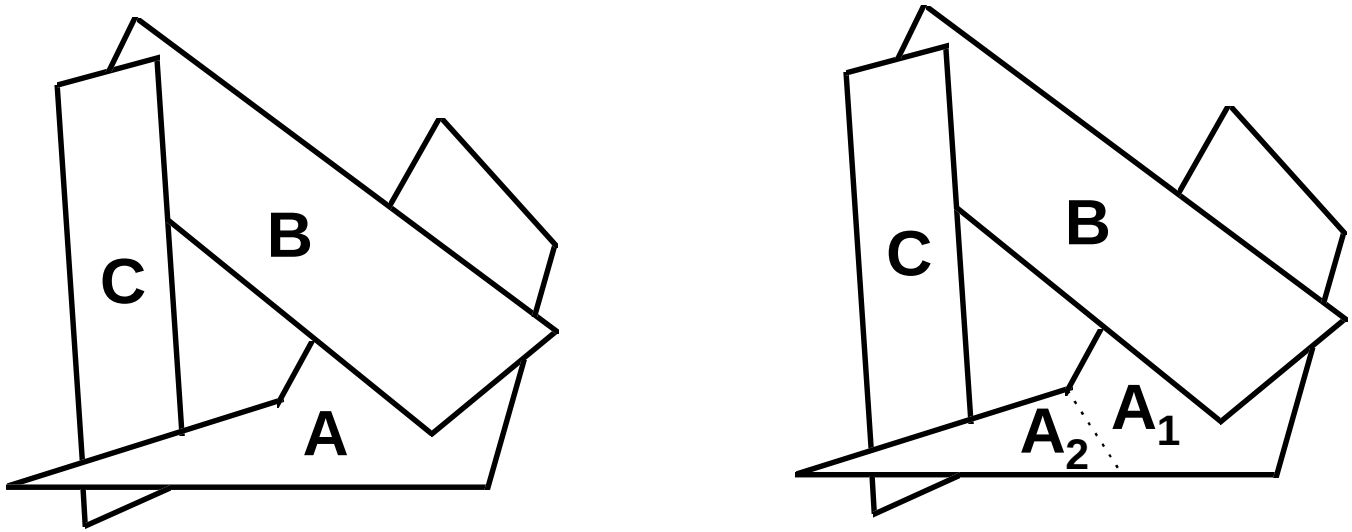
- jestliže neexistuje průsečík hran ***P*** a ***Q***, je třeba ještě zkontrolovat, zda neleží ploška ***P*** **celá uvnitř** ***Q*** nebo naopak. V takovém případě opět porovnáme souřadnice ***z***



2. fáze: změna pořadí

- jestliže nelze z nějakého důvodu nakreslit ***P*** před ***Q***, zkusíme přesunout plošku ***Q*** **na začátek seznamu *S*** (ještě před P)
  - pro ***Q*** budeme opět provádět všechny testy 2. fáze (jak jsme je popsali s ploškou ***P***)
  - testy nového kandidáta ***Q*** proti ***P*** už byly z velké části provedeny, stačí pouze doplnit obrácené testy **B** a **C**
- kvůli **zacyklení** se musí každý kandidát označit zvláštním příznakem

2. fáze: zacyklení



- jestliže je testován některý kandidát podruhé, došlo k **zacyklení**
- cyklus lze odstranit **rozštěpením** některé plošky (správné pořadí je pak $\mathbf{A_1}$, **B**, **C**, $\mathbf{A_2}$)

Konec

Další informace:

- **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
- **LAN** na Malé Straně:
  - **barbora\usr:\vyuka\pelikan\6**