Geometria – poznámky z prednášky

Sylabus, resp. čo treba vedieť na skúške

- *Pojem* skalárny súčin *a* vektorový súčin.
- *Algoritmus na spočítanie* **obsahu všeobecného n-uholníka v 2D** *v* $O(n)$
- *Algoritmus na spočítanie* **konvexného obal** n **bodov v 2D** *v* $O(n \log n)$. *Stačí vedieť jeden z algoritmov.*

Na skúške by ste mali vedieť vysvetliť ako fungujú tieto algoritmy, prečo fungujú, prečo majú takú časovú zložitosť, ako majú.

Body a vektory

Bod aj vektor reprezentujeme ako dvojicu čísel – x-ovú a y-ovú súradnicu. Na týchto dvojiciach môžeme definovať dodatočné operácie, ako napríklad sčítanie a odčítanie.

$$(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2)$$
$$(x_1, y_1) - (x_2, y_2) = (x_1 - x_2, y_1 - y_2)$$

Platí pri tom, že súčet dvoch vektorov je vektor, rozdiel dvoch vektorov je vektor, súčet bodu a vektoru je bod a rozdiel dvoch bodov je vektor.

Napríklad $A - B$ je vektor, ktorý vedie z bodu B do bodu A.

Skalárny súčin

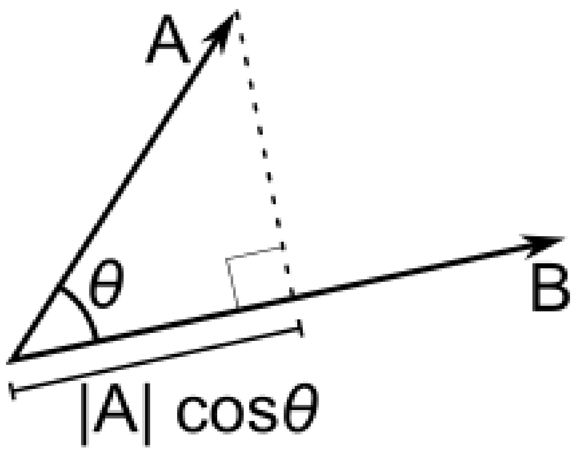
Skalárny súčin (angl. dot product) je operácia na dvoch n-rozmarných vektoroch, ktorú poznáme už z algebry. Výsledkom tejto operácie je číslo (čiže skalár a nie vektor). Skalárny súčin označujeme bodkou. Nech $U = (u_1, \ldots, u_n)$ a $V = (v_1, \ldots, v_n)$. Potom

$$U \cdot V = u_1 v_1 + u_2 v_2 + \cdots + u_n v_n$$

Zároveň, keď θ je uhol, ktorý zvierajú vektory, tak platí

$$U \cdot V = ||U|| \cdot ||V|| \cdot \cos\theta$$

kde $||X||$ označuje dĺžku vektora X.



Skalárnym súčinom vieme zisťovať dĺžku vektora ($U \cdot U = ||U||^2$), vieme overovať, či sú dva vektory kolmé (keď $U \cdot V = 0$), rozhodovať, či kolineárne vektory ukazujú rovnakým alebo opačným smerom (podľa znamienka skalárneho súčinu). Neskôr si ukážeme, ako využiť skalárny súčet pri výpočte vzdialenosti bodu od priamky alebo na nájdenie päty kolmice na danú priamku vedúcu cez daný bod.

Skalárny súčin má mnohé využitia v strojovom učení aj ďalších oblastiach matematiky, fyziky a informatiky.

Príklad 1.a

Máme body A, B, C ležiace na jednej priamke. Zistite, či bod C leží na úsečke AB.

Riešenie 1.a

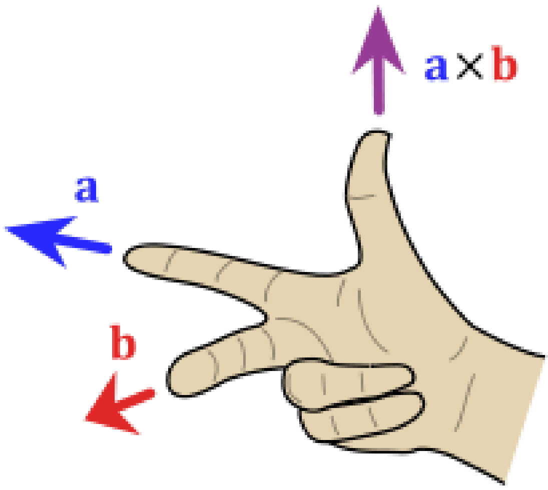
Spočítame $x = (A - C) \cdot (B - C)$. Ak nám vyjde záporné x, bod C leží vo vnútri úsečky. Ak kladné, bod leží mimo úsečky. Ak je x nula, C je totožný s jedným z bodov A, B.

Vektorový súčin

Na rozdiel od skalárneho súčinu, vektorový súčin (angl. cross product) je definovaný len v troch rozmeroch. Matematická definícia je

$$(u_1, u_2, u_3) \times (v_1, v_2, v_3) = (u_2v_3 - u_3v_2, u_3v_1 - u_1v_3, u_1v_2 - u_2v_1)$$

Výsledkom vektorového súčinu dvoch vektorov je vektor, ktorý je kolmý na oba vektory a jeho dĺžka je rovná obsahu štvoruholníka $((0, 0), U, U+V, V)$. (Existujú dva opačné vektory s takouto dĺžkou, ak chceme zistiť, ktorý z nich je správny, použijeme pravidlo pravej ruky. Ukazovák ukazuje v smere prvého vektory, prostredník v smere druhého a palec bude ukazovať v smere vektorového súčinu.)



Špeciálne ak sú pôvodné vektory lineárne závislé, je výsledkom vektorového súčinu nula (t.j. vektor $(0, 0, 0)$).

Pre nás je najdôležitejší vektorový súčin v dvoch rozmeroch, teda pre $u_3 = v_3 = 0$. Potom $U \times V = (0, 0, u_1v_2 - u_2v_1)$. Zaujímavá je orientovaná dĺžka výsledného vektora, čiže $u_1v_2 - u_2v_1$. Keď budeme hovoriť o vektorovom súčine ako o čísle, budeme mať namysli hodnotu $u_1v_2 - u_2v_1$.

Na čo je teda vektorový súčin dobrý? Opäť ním dokážeme zistiť niečo o uhle medzi vektormi.

$$||U \times V|| = ||U|| \cdot ||V|| \cdot \sin\theta$$

Ak sú vektory kolineárne (ukazujú rovnakým alebo opačným smerom), je výsledok 0. Ak je V „naľavo“ od U, je ich vektorový súčin kladný, a ak je V „napravo“ od U, je ich vektorový súčin záporný.

Vďaka tomu vieme zisťovať, či daný bod leží na danej priamke, či je bod vo vnútri konvexného n-uholníka, či sa dve úsečky pretínajú, či je bod vo vnútri ľubovoľného n-uholníka a mnoho ďalších vecí.

Príklad 2.a

Máme daný bod P a priamku prechádzajúcu bodmi AB. Zistite, či sa bod nachádza na priamke, naľavo od nej alebo napravo od nej (keď sa pozeráme v smere z A do B).

Príklad 2.b

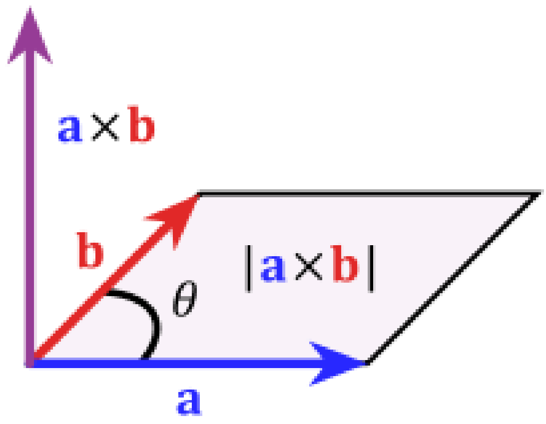
Máme daný bod P a konvexný n-uholník s vrcholmi $A_1, \ldots, A_n$. Zistite, či je bod vo vnútri n-uholníka.

Príklad 2.c

Máme dané dve úsečky AB a CD. Zistite, či sa pretínajú.

Príklad 2.d

Máme daný bod P a ľubovoľný n-uholník s vrcholmi $A_1, \ldots, A_n$. Zistite, či je bod vo vnútri n-uholníka.



Riešenie 2.a

Spočítame $(B - A) \times (P - A)$. Ak vyjde 0, bod leží na priamke. Ak vyjde kladné číslo, bod P je naľavo, ak vyjde záporné číslo bod je napravo.

Riešenie 2.b

Spočítame $x_i = (A_{i+1} - A_i) \times (P - A_i)$ (pričom $A_{n+1} = A_1$). Ak sú všetky hodnoty kladné, alebo všetky záporné, bod P je vo vnútri.

Riešenie 2.c

Overíme, či úsečka AB pretína priamku CD a úsečka CD pretína priamku AB. Overenie prvej podmienky spravíme napríklad tak, že sa pozrieme na vektorové súčiny $(A - C) \times (D - C)$ a $(B - C) \times (D - C)$. Tieto musia mať opačné znamienka, resp. ich súčin musí byť záporný.

Riešenie 2.d

Ak je bod P vo vnútri, každá polpriamka, ktorá v ňom začína pretína n-uholník nepárny počet krát. Ak je P vonku, tak je počet pretnutí vždy párny. Takže počítame spočítame priesečníky nejakej polpriamky s hranami n-uholíka. Treba si však dávať pozor na prípady, kedy vrcholy n-uholníka ležia na priamke.

Zatiaľ sme sa pozerali len na znamienko vektorového súčinu. Keď sa však pozrieme na jeho hodnotu (dĺžku vektora) budeme schopní spočítať mnoho ďalších vecí súvisiacich s obsahmi.

Príklad 2.e – Obsah n-uholníka

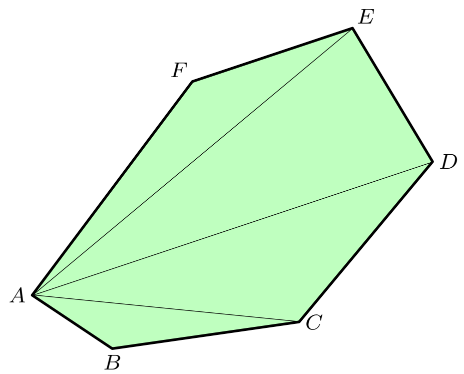
Máte zadaný n-uholník. Spočítajte jeho obsah.

Príklad 2.f

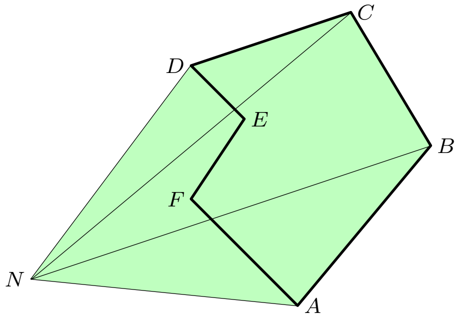
Máte zadaný n-uholník. Spočítajte, kde sa nachádza jeho ťažisko.

Riešenie 2.e – Obsah n-uholníka

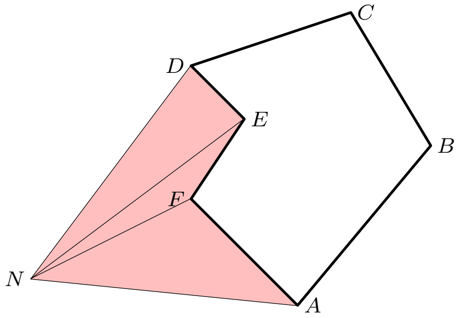
Pomocou vektorového súčinu dokážeme ľahko spočítať obsah trojuholníka ABC. Je to $|(B-A)\times(C-A)|/2$. Vďaka tomuto vzťahu vieme spočítať aj obsah konvexného n-uholníka. Stačí si n-uholník rozdeliť na trojuholníky a sčítať ich obsahy.



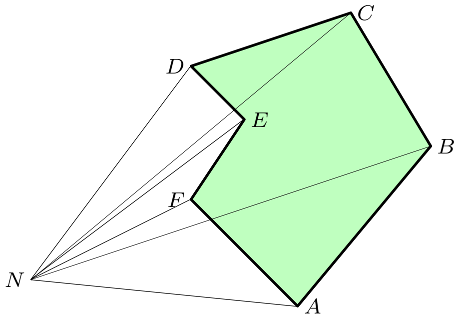
Pokiaľ z výrazu vynecháme absolútnu hodnotu, dostaneme takzvaný orientovaný obsah. Jeho znamienko bude kladné, ak body útvaru boli udané proti smeru hodinových ručičiek a záporné v opačnom prípade. Vďaka tomu dokážeme spočítať obsah ľubovoľného, aj nekonvexného n-uholníka. Zatiaľ predpokladajme, že vrcholy n-uholníka máme zadané v poradí proti smeru hodinových ručičiek. Ukážeme si to na obrázkoch 1, 2 a 3. N môže byť ľubovoľný bod, väčšinou sa kvôli pohodlnosti vyberá bod $(0, 0)$.



Obr. 1: Obsahy trojuholníkov NAB, NBC, NCD.



Obr. 2: Záporné obsahy trojuholníkov NDE, NEF, NFA.



Obr. 3: Výsledný obsah.

Predstavme si, že sme hrany n-uholníka natreli zvnútra zelenou farbou a zvonka červenou. Keď sa pozrieme na n-uholník s ľubovoľného bodu N, niektoré hrany uvidíme z červenej strany a niektoré so zelenej. Teraz si potrebujeme uvedomiť dve pozorovania (stále predpokladáme, že vrcholy n-uholníka máme zadané v protismere hodinových ručičiek).

- Hranu AB vidíme zo zelenej strany, pokiaľ vektorový súčin $(A - N) \times (B - N)$ je kladný a z červenej strany pokiaľ je záporný.
- Keď sčítame obsahy trojuholníkov NXY, kde XY je zelená hrana n-uholníka, a odčítame obsahy trojuholníkov NXY, kde XY je červená hrana n-uholníka, dostaneme obsah n-uholníka.

Dôkazy oboch pozorovaní si rozmyslite sami. Tieto tvrdenia následne môžeme využiť na vyriešenie úlohy. Vo všeobecnosti spočítame obsah n-uholníka nasledovne. Predstavíme si každý vrchol n-uholníka ako vektor A_i vedúci z bodu $(0, 0)$ do daného vrchola. Potom obsah n-uholníka je

$$\frac{\left|\left|\sum_{i=1}^{n} A_{i+1} \times A_i\right|\right|}{2}$$

pričom $A_{n+1} = A_1$

Vo výpočte z výsledného obsahu spočítame absolútnu hodnotu. Totiž ak by sme mali zadané vrcholy n-uholníka v smere hodinových ručičiek, vyšiel by nám výsledný obsah záporný.

Riešenie 2.f

Ťažisko trojuholníka vieme spočítať (aritmetický priemer jeho vrcholov). Vezmeme všetky trojuholníky, ktorých obsahy sme počítali v predošlom cvičení, každému spočítame ťažisko a výsledok bude váhovaný priemer týchto ťažísk. Váha jedného ťažiska bude orientovaný obsah trojuholníka.

Konvexný obal

Konvexný obal množiny A je prienik všetkých konvexných nadmnožín A. Špeciálne konvexný obal konečnej množiny bodov v rovine je konvexný k-uholník. Zároveň je konvexný obal n-bodov útvar s najmenším obvodom, ktorý obsahuje všetkých n bodov.

Tento n-uholník dokážeme nájsť v čase $\Theta(n \log n)$. (Ukazovali sme si, že vo všeobecnosti sa konvexný obal n bodov nedá nájsť v čase lepšom ako triedenie n čísel.)

Na prednáške sme si ukazovali dva algoritmy. Algoritmy fungujú na princípe dynamického programovania, t.j. snažia sa najprv spočítať konvexný obal jedného bodu, potom dvoch bodov, troch bodov, atď, až kým nenájdu konvexný obal celej množiny.

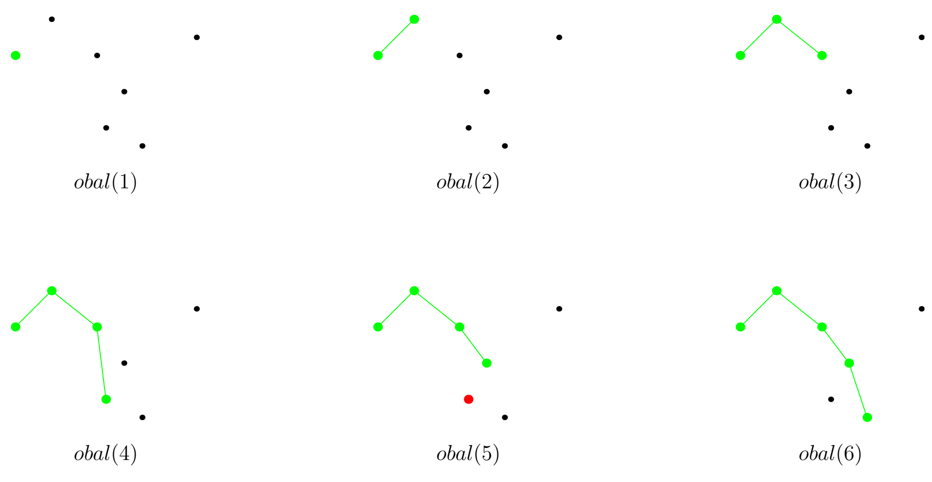
Horný a dolný obal

Tento algoritmus nájde osobitne horný obal všetkých bodov, teda najnižšiu konkávnu lomenú čiaru, pod ktorou sa nachádzajú všetky body. Podobne dolný obal bude najvyššia konvexná lomená čiara, nad ktorou sa nachádzajú všetky body.

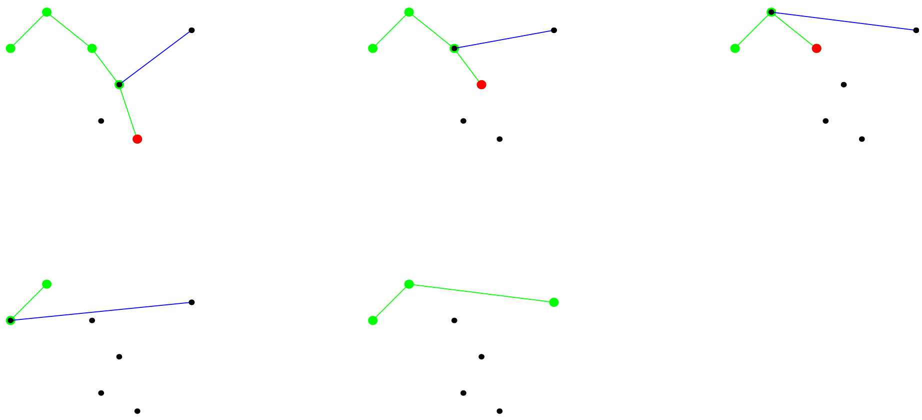
Horný obal nájdeme takto. Vytvoríme si prázdny zoznam B, v ktorom na konci chceme mať horný konvexný obal. Utriedime všetky body, ktorým hľadáme konvexný obal, zľava doprava (t.j. prioritne podľa x-ovej súradnice a v prípade remízy podľa y-ovej). Označíme si utriedenú postupnosť bodov $A_1, \ldots, A_n$.

Teraz budeme postupne počítať horný obal najľavejších k bodov pre $k \in \{1..n\}$. Každý z týchto obalov je nejaká lomená čiara, ktorá spája niektoré z prvých k vrcholov. Postupnosť vrcholov na obale prvých k bodov budeme označovať $obal(k)$.

obal(k) vždy obsahuje A_k. $obal(1) = \{A_1\}$. Pre $k > 1$, obal(k) obsahuje nejaký začiatok postupnosti $obal(k-1)$ a potom prvok A_k. obal(k) dokážeme skonštruovať tak, že zoberieme $B_1, \ldots, B_h$ postupnosť bodov predošlého obalu a skontrolujeme, či sa B_h nenachádza pod $B_{h-1}A_k$. Ak áno, B_h zmažeme z postupnosti a overíme, či sa B_{h-1} nenachádza pod úsečkou $B_{h-2}A_k$. A tak pokračujeme, až kým nám neostane v B len jeden vrchol, alebo kým sa posledný bod B nebude nachádzať nad úsečkou spájajúcou predposledný bod B a A_k.



Obr. 4: Horné obaly prvých k bodov pre $k \in \{1..6\}$



obal(7)

Obr. 5: Počítanie obalu pre $k = 7$. Zoberieme $obal(6)$ a postupne odstraňujeme body z konca, ktoré sa nachádzajú pod modrou úsečkou.

Na určovanie toho, či sa bod nachádza nad úsečkou alebo pod úsečkou použijeme vektorový súčin. Y neleží pod XZ, pokiaľ $(Y - X) \times (Z - X) < 0$.

Algoritmus na spočítanie horného obalu bude teda fungovať nasledovne. Postupne prejdeme všetky body a pridáme ich do konvexného obalu B. Avšak vždy pred tým, ako pridáme bod A_i do obalu, vyhodíme všetky body, ktoré sa nachádzajú pod horným obalom bodov $A_1, \ldots, A_i$. Dolný obal C nájdeme veľmi podobne, stačí prejsť body v A v opačnom poradí.

```
# v A máme vzostupne utriedené body podľa x-ovej súradnice
B,C = [],[]

for a in A:
    while len(B) > 1 and cross_product(B[-1]-B[-2], a-B[-2]) >= 0:
        B.pop()
    B.append(a)

for a in reversed(A):
    while len(C) > 1 and cross_product(C[-1]-C[-2], a-C[-2]) >= 0:
        C.pop()
    C.append(a)
```

Celý konvexný obal dostaneme tak, že zreťazíme zoznamy B a C bez posledných prvkov. Takto dostaneme zoznam vrcholov na konvexnom obale vymenovaných v smere hodinových ručičiek (takže by sa to v skutočnosti malo volať konkávny obal?).

Časová zložitosť je $O(n \log n)$, lebo sme body utriedili. Ak by sme už mali body utriedené, nájdeme konvexný obal v čase $O(n)$. (Každý bod najviac raz pridáme do obalu a teda ho najviac raz odoberieme.) Vo všeobecnosti nevieme hľadať konvexný obal rýchlejšie ako triediť. Totiž ak máme pole čísel $a_1, \ldots a_n$ tak konvexný obal bodov $(a_1, a_1^2), (a_2, a_2^2), \ldots (a_n, a_n^2)$ je tvorený všetkými prvkami v utriedenom poradí.

Graham scan

Druhý algoritmus funguje veľmi podobne, avšak konštruuje celý konvexný obal naraz. Najprv nájdeme najľavejší bod (v prípade remízy najspodnejší z nich), o ktorom vieme, že sa určite nachádza na konvexnom obale. Označme tento bod A_0.

Následne utriedime ostatné body $A_1, \ldots, A_{n-1}$ podľa uhla vektora $A_{n-1} - A_0$ s osou y. (Môžeme použiť triedenie porovnávaním a namiesto porovnávania sa budeme pozerať vždy na znamienko vektorového súčinu porovnávaných vektorov.)

Následne úplne rovnakým spôsobom prejdeme tieto utriedené body.

```
# v A máme vzostupne utriedené body podľa uhla,
# špeciálne najľavejší bod je na prvej pozícii zoznamu
```

```
B = []

for a in A:
    while len(B) > 1 and cross_product(B[-1]-B[-2], a-B[-2]) >= 0:
        B.pop()
    B.append(a)
```

Priamky

Priamky sa dajú v programoch reprezentovať rôznymi spôsobmi.

- dvojica rôznych bodov A, B určuje priamku, ktorá vedie cez oba tieto body
- dvojica bod a nenulový vektor, A, U určuje priamku, ktorá vedie cez body A a $(A+U)$
- dvojica reálnych čísel (a, b) popisuje lineárnu funkciu $f(x) = ax + b$, ktorá má tvar priamky. Keďže zvislá priamka sa týmto spôsobom nedá popísať, väčšinou sa táto reprezentácia nepoužíva
- trojica reálnych čísel (a, b, c) popisuje rovnicu $ax + by + c = 0$, ktorej riešením sú body (x, y) ležiace na priamke. Zaujímavé je, že vektor $V = (a, b)$ je kolmý na priamku (a, b, c).

V tejto časti si ukážeme ako spočítať vzdialenosť bodu od priamky, nájsť najbližší bod priamky k danému bodu a ukážeme si, ako nájsť priesečník dvoch priamok. Existuje množstvo spôsobov, ako tieto veci spočítať, väčšina z nich je veľmi komplikovaná a vyžaduje rozoberanie okrajových prípadov, my si však ukážeme také postupy, ktoré sú elegantné a jednoduché. Pre tieto účely sa najviac hodí reprezentácia pomocou bodu a vektora.

Príklad 4.a

Majme priamku A, U (prechádzajúcu bodmi A a (A+U)) a bod P. Nájdite bod priamky, ktorý je najbližší k bodu P. Tento bod sa volá päta kolmice vedúcej cez P.

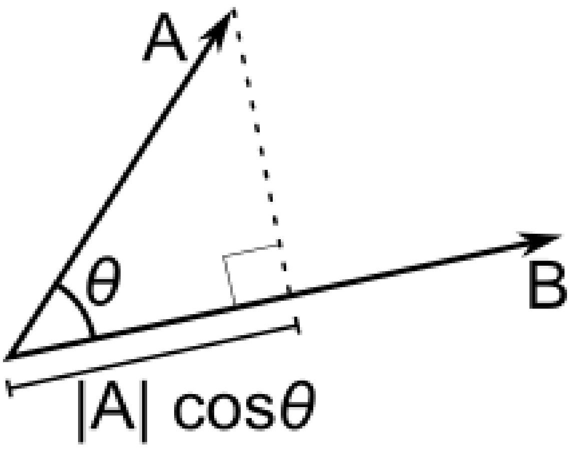
Príklad 4.b

Majme priamku A, U (prechádzajúcu bodmi A a (A+U)) a bod P. Zistite vzdialenosť bodu P od priamky.

Príklad 4.c

Majme dve priamky A, U a B, V. Nájdite ich priesečník.

Riešenie 4.a



Pozrime sa na „tieň“ ktorý vrhá vektor $(P - A)$ na priamku. Na konci tohto tieňa je hľadaný bod. Dĺžka tieňa je

$$t = \frac{(P - A) \cdot U}{||U||}$$

Hľadaný bod je na pozícii $A + \frac{t}{||U||}U$. Po rozpísaní dostaneme

$$A + \frac{(P - A) \cdot U}{U \cdot U}U$$

Riešenie 4.b

Jeden spôsob je spočítať vzdialenosť bodu P a päty kolmice vedúcej cez P. Existuje však aj jednoduchší spôsob. Stačí si uvedomiť, že $||(P-A) \times U||$ je obsah kosodlžníka, ktorého podstava je dlhá ——U—— a výška je vzdialenosť bodu P od priamky.

Tým pádom vzdialenosť P od priamky spočítame ako

$$\frac{||(P-A) \times U||}{||U||}$$

Riešenie 4.c

Všetky body ležiace na prvej priamke majú tvar $A + tU$ pre nejaké reálne číslo k. Podobne body ležiace na druhej priamke majú tvar $B + sV$.

Pre priesečník priamok teda platí

$$A + tU = B + sV$$
$$tU - sV = B - A$$

$$tU_x - sV_x = B_x - A_x$$
$$tU_y - sV_y = B_y - A_y$$

Vynásobíme prvú rovnicu V_y, druhú $-V_x$ a sčítame ich.

$$t(U_xV_y - U_yV_x = B_xV_y - B_yV_x - (A_xV_y - A_yV_x)$$

Predstavme si body A,B ako vektory vedúce z (0,0) do daných bodov.

$$t(U \times V) = B \times V - A \times V$$
$$t = \frac{B \times V - A \times V}{U \times V}$$

Keď poznáme t, nájdeme priesečník priamok ako $A + tU$. Všimnime si, že pokiaľ $t \in \langle 0, 1\rangle$ tak priesečník leží na úsečke z A do $(A + U)$. Ľahko by sme zovšeobecnili postup na hľadanie priesečníkov úsečiek.

Tiež si všimnime, že $U \times V$ je nula, ak sú priamky rovnobežné. Vtedy priesečník nie je definovaný.