

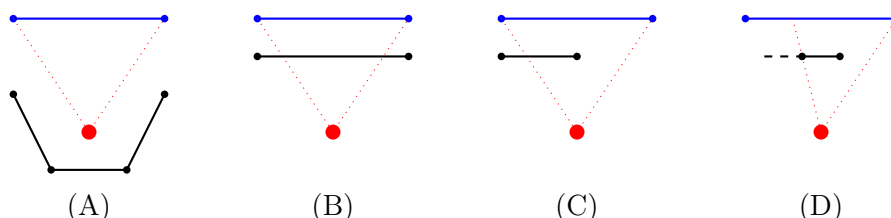
## 1. Aed (aed)

1 sek / 3 sek

100 punkti

Tasandil on  $N$  tipuga hulknurk ja punkt  $U$ , mis võib olla nii hulknurga sees kui sellest väljas. Leida, milliseid hulknurga külgi punktis  $U$  olev vaatleja näeb.

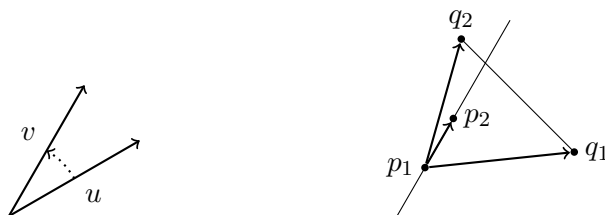
Ülesande lahendamiseks võime vaadelda hulknurga iga külge (alloleval joonisel sinine) eraldi ja uurida, kas mingi osa sellest on vaatlejale nähtav. Nähtavat osa võime arvestada sektorina (punane), mida hulknurga ülejäänud külgi (mustad) järjest läbi vaadates jooksvalt kitsendame.



Kui hulknurga ülejäänud külgi läbida vaadeldava külje ühest otsast alates ja mööda hulknurga piirjoont edasi liikudes, siis võib iga järgmine külg olla kas täiesti väljaspool vaatesektorit (A), läbida sektori (B) või olla sektoris ainult ühte otsa pidi (C).

Oluline on, et varjav külg ei saa olla tervenisti sektori sisemuses, sest siis pidi ka sellele eelnev külg vähemalt osaliselt sektori sees olema ja oleks seda kitsendanud (D). See tähendab, et nähtav sektor püsib kogu protsessi vältel sidusana ja me saame selle üle arvet pidada kahe kiirega.

Kiire ja lõigu vastastikuse asendi tuvastamiseks võime kasutada vektorkorrutise omadusi. Tasandi vektorite  $\vec{u} = (x_u, y_u)$  ja  $\vec{v} = (x_v, y_v)$  vektorkorrutis  $\vec{u} \times \vec{v} = x_u \cdot y_v - x_v \cdot y_u$  on positiivne, kui pööre vektori  $\vec{u}$  suunalt vektori  $\vec{v}$  suunale on vastupäeva (alloleval joonisel vasakul), ja negatiivne, kui pööre on päripäeva.<sup>1</sup> See tähendab, et punkte  $p_1$  ja  $p_2$  läbiv sirge lõikab punkte  $q_1$  ja  $q_2$  ühendavat lõiku parajasti siis, kui  $p_1\vec{p}_2 \times p_1\vec{q}_1$  ja  $p_1\vec{p}_2 \times p_1\vec{q}_2$  on erimärgilised (joonisel paremal).



Lisaks vaatesihiga lõikumise faktile on vaja teada ka seda, kas potentsiaalselt varjav lõik on vaatleja ja vaadeldava lõigu vahel. Lõikepunkti kauguse punktist  $p_1$  saame avaldada järgmistest tähelepanekutest.

Kõik punkte  $p_1$  ja  $p_2$  läbival sirgel asuvad punktid  $p$  avalduvad kujul  $p = p_1 + t \cdot p_1\vec{p}_2$ . Samamoodi avalduvad sirge  $q_1q_2$  punktid kujul  $q_1 + u \cdot q_1\vec{q}_2$ . Lõikepunkt peab rahuldama mõlemat võrrandit. Seega otsime  $t$  ja  $u$  väärtusi, mille korral

$$p_1 + t \cdot p_1\vec{p}_2 = q_1 + u \cdot q_1\vec{q}_2$$

<sup>1</sup>Tegelikult on vektorkorrutis ruumivektorite vahel defineeritud tehe, mille tulemus on samuti ruumivektor. Aga kuna tasandivektorite vektorkorrutise  $x$ - ja  $y$ -koordinaadid on alati nullid, siis kasutatakse tasandi geomeetria ülesannetes vektorkorrutise asemel selle  $z$ -koordinaadi väärtust.

ehk koordinaatide kaupa välja kirjutades

$$\begin{cases} x_{p_1} + t \cdot (x_{p_2} - x_{p_1}) = x_{q_1} + u \cdot (x_{q_2} - x_{q_1}) \\ y_{p_1} + t \cdot (y_{p_2} - y_{p_1}) = y_{q_1} + u \cdot (y_{q_2} - y_{q_1}) \end{cases},$$

millest esimest võrrandit  $(y_{q_2} - y_{q_1})$ -ga ja teist võrrandit  $(x_{q_2} - x_{q_1})$ -ga korrutades, neid üksteisest lahutades ja siis sarnaseid liikmeid koondades saame avaldada

$$t = \frac{(x_{q_1} - x_{p_1}) \cdot (y_{q_2} - y_{q_1}) - (y_{q_1} - y_{p_1}) \cdot (x_{q_2} - x_{q_1})}{(x_{p_2} - x_{p_1}) \cdot (y_{q_2} - y_{q_1}) - (y_{p_2} - y_{p_1}) \cdot (x_{q_2} - x_{q_1})} = \frac{p_1 \vec{q}_1 \times q_1 \vec{q}_2}{p_1 \vec{p}_2 \times q_1 \vec{q}_2}. \quad (1)$$

Nüüd saaksime avaldada lõikepunkti kauguse punktist  $p_1$  kujul  $d = t \cdot |p_1 p_2|$ , aga seda pole tegelikult vaja. Piisab tähelepanekust, et  $t < 0$  vastab punktidele, mis on sirgel punktist  $p_2$  vaadates teisel pool punkti  $p_1$ ,  $0 < t < 1$  vastab punktidele, mis on  $p_1$  ja  $p_2$  vahel, ning  $t > 1$  vastab punktidele, mis on punktist  $p_1$  vaadates teisel pool punkti  $p_2$ . Sisuliselt annab  $t$  väärtus meile lõikepunkti koordinaadi teljel, kus  $p_1 \vec{p}_2$  on ühikvektor, ja me võime erinevate lõikepunktide omavahelisi asendeid uurida neile vastavate  $t$  väärtuste võrdlemise teel.

Absoluutarvudes kauguste asemel  $t$  väärtuste kasutamisel on üks põhimõtteline eelis: kui punktide koordinaadid on täisarvud, siis on valemist (1) saadud  $t$  väärtused alati ratsionaalarvud, mida võime programmis hoida harilike murdudena ja mille võrdlemine on sel juhul täpne, ümardamisvigadeta. Nii ongi tehtud Pythoni lahenduses failis `sol.py`.

Praktiliselt on selles ülesandes võimalik saada kõigis testides õiged vastused ka topelttäpsusega reaalarve kasutades, nagu on tehtud C++ lahenduses failis `sol.cpp`, kus valenegatiivsete tulemuste vältimiseks võrreldakse arve teatud täpsuspiirini: üksteisest vähem kui  $10^{-6}$  võrra erinevad arvud loetakse võrdseks.

Pythoni harilike murdudega lahendust C++ ümber kirjutades peaks olema ettevaatlik. Nimelt võivad murdudega arvutades nende nimetajad ja lugejad C++ piiratud suurusega (32 või 64 bitti) täisarvutüüpides ületäitumisi põhjustada. Pythonis seda ohtu ei ole, sest Python läheb väärtuste kasvades automaatselt üle ülipikkade täisarvude kasutamisele.

Teiselt poolt väärub mainimist, et nii Pythoni kui C++ täisarvudes ja ka Pythoni ujukomaarvudes põhjustab nulliga jagamise katse alati veaolukorra, aga C++ ujukomaarvudes annab positiivse või negatiivse arvu nulliga jagamine tulemuseks vastavalt  $+\infty$  või  $-\infty$ .<sup>2</sup> Nende ebanormaalsete väärtustega edasi arvutades peab küll ettevaatlik olema, aga siiski võimaldab see sageli koodi märgatavalt lihtsustada.

---

<sup>2</sup>Selline käitumine ei ole C++ standardi järgi kohustuslik. See on aga lubatud laiendus ja olümpiaadi serveris lahenduste testimiseks kasutatav kompilaator nii käitub.