1. Õunte jagamine (`jaga`) 1 sek / 3 sek 20 punkti

On antud N elemendiga massiiv K. Kust teha massiiv pooleks nii, et vasaku ja parema poole summad oleks “võimalikult võrdsed”, s.t. et poolte vahe absoluutväärtus oleks võimalikult väike? ($1 \leqslant N \leqslant 10^6$)

Lahendus

Arvutame iga i jaoks $V_i = K_1 + K_2 + \cdots + K_i$ ja $P_i = K_{i+1} + K_{i+2} + \cdot + K_N$. Vastus on see i väärtus, mille korral $|V_i - P_i|$ on vähim.

Selline naiivne algoritm annab küll õige vastuse, aga kulutab iga i jaoks summade arvutamiseks kokku umbes N liitmist. Kuna võimalikke i väärtusi on N, kulub kokku umbes N^2 tehet, mida on suuremates testides ajalimiidi sisse mahtumiseks liiga palju.

Töömahtu saame vähendada, kui paneme tähele, et V_i ja P_i teades pole vaja V_{i+1} ja P_{i+1} arvutamist otsast peale alustada. Selle asemel saame arvutada $V_{i+1} = V_i + K_{i+1}$ ja $P_{i+1} = P_i - K_{i+1}$. Sel moel saame $i = 1$ jaoks algseisu arvutada umbes N tehtega ja edasi iga järgmise i jaoks uue seisu arvutada vanast vaid kahe tehtega. Seega kulub kõigi N võimaliku i väärtuse uurimiseks kokku ainult umbes $3N$ tehet.

2. Seoste järeldamine (`seos`) 1 sek 30 punkti

Meil on kolm muutujat a, b, c ja teame kaht fakti, millest kumbki on ühel järgmistest kujudest: $x < y$, $x \leqslant y$, $x = y$, $x \geqslant y$, $x > y$. Koosta 3×3 tabel, kus igale muutujale vastab üks rida ja üks veerg ning igasse ruutu on kirjutatud kõige tugevam sisendist järelduv seos nende kahe muutuja vahel.

Ülesandel on kaks radikaalselt erinevat lähenemisviisi.

Lahendus 1: palju `if`-lauseid

Esiteks eemaldame sisendist võrdusseosed. Kui sisendis on tingimus $x = y$, siis asendame sisendis igal pool muutuja y muutujaga x ning viskame selle tingimuse minema. Pärast kopeerime tabelis vastavad read ja veerud.

Teiseks eemaldame seosed, mis võrdlevad mingit muutujat iseendaga. Kui sisendis on tingimus $x < x$ või $x > x$, siis see on vastuolu. Kui sisendis on tingimus $x \leqslant x$ või $x \geqslant x$, siis see on kasutu informatsioon.

Nüüd on meil kuni kaks võrratust. On mõned erinevad juhud.

- *Pole ühtegi allesjäänud võrratust.*
- *Allesjäävad võrratused mainivad ainult kahte muutujat.* Olgu nad x ja y.
  - *Allesjäävad võrratused on vastuolulised.*
  - *Allesjäävad võrratused on $x \leqslant y$ ja $x \geqslant y$.* Siit järeldub, et $x = y$; täidame tabeli vastavalt.
  - *Ülejäänud juhud.* Kui siin on kaks võrratust, siis valime nendest tugevama. Alles jääb üks võrratus; täidame tabeli vastavalt.
- *Allesjäävad võrratused mainivad kolme muutujat.* Täpselt ühte nendest mainitakse siis kaks korda, olgu see x; ülejäänud kaks olgu y ja z.

  - *y ja z asuvad arvteljel samal pool arvu x.* Siis me ei saa võrratustest y ja z omavahelise seose kohta midagi järeldada ehk tabelisse läheb `??`.
  - *y ja z asuvad arvteljel teine seisel pool arvu x.* Konkreetsuse mõttes oletame, et $y \leqslant x$ või $y < x$ ja $x \leqslant z$ või $x < z$.
    * *Mõlemad võrratused sisaldavad võrdusmärki.* Siis saame järeldada, et $y \leqslant z$, aga mitte $y < z$; täidame tabeli vastavalt.
    * *Vähemalt üks võrratus on range.* Siis saame järeldada, et $y < z$; täidame tabeli vastavalt.

Keerukus $\mathcal{O}(1)$.

Lahendus 2: proovime kõiki kolmikuid

Teeme kolm 3×3 tabelit L, G ja E. Eesmärk on, et nad näitaksid järgmiseid asju.

- $L[i][j]$ näitab, kas on võimalik, et i-s muutuja on väiksem kui j-s muutuja;
- $E[i][j]$ näitab, kas on võimalik, et i-s muutuja on võrdne j-nda muutujaga;
- $G[i][j]$ näitab, kas on võimalik, et i-s muutuja on suurem kui j-s muutuja.

Edasi teeme järgmist.

- Käime läbi kõik võimalikud arvude kolmikud (a, b, c).
- Kontrollime, kas antud kolmik (a, b, c) vastab antud tingimustele.
  - Kui jah, siis uuendame tabeleid, nt. kui $a < c$, siis märgime, et $L[0][2] = 1$.

Erinevaid kolmikuid (a, b, c) on muidugi lõpmatu arv. Aga on selge, et piisab, kui vaatame ainult kolmikuid, kus $a, b, c \in \{0, 1, 2\}$.

Kui ükski kolmik ei vastanud tingimustele, siis on sisend vastuolu. Vastasel juhul saame tabelitest L, E, G vastuse kokku panna.

Keerukus $\mathcal{O}(1)$.

3. Superratsu (`ratsu`) — 1 sek — 40 punkti

$R \times V$ ruudustikul on mõned ruudud blokeeritud. Ratsul on vaja liikuda ruudust a ruutu b minimaalse arvu käikudega. Ratsu käib, nagu ratsu ikka, aga ühe korra võib ta teha *superkäigu*: liikuda kolm ruutu horisontaalselt ja ühe vertikaalselt (või vastupidi).

Leia lühim tee a ja b vahel. ($1 \leqslant R, V \leqslant 100$)

Lahendus

Olgu ratsu algusruut (r_s, v_s) ja lõppruut (r_f, v_f).

Mistahes hetkel saame kirjeldada ratsu "olekut" kolmikuna (r, v, s). Siin:

- r on rea number, millel ratsu on;
- v on veeru number, millel ratsu on;
- s on kas 0 või 1: näitab, kas ratsu on superkäigu juba ära teinud.

Ühe käiguga võib ratsu olek muutuda järgmiselt.

- $(r, v, s) \to (r+2, v+1, s)$;
- $(r, v, s) \to (r+2, v-1, s)$;
- $(r, v, s) \to (r+1, v+2, s)$;
- jne;
- $(r, v, 0) \to (r+3, v+1, 1)$;
- $(r, v, 0) \to (r+3, v-1, 1)$;
- $(r, v, 0) \to (r+1, v+3, 1)$;
- jne.

Need muutused saavad toimuda ainult siis, kui mõlemad ruudud jäävad laua peale ega ole blokeeritud.

Joonistame graafi, kus on üks tipp iga oleku kohta ning sellised suunatud servad, nagu ülal toodud. Peame leidma järgmised lühimad teed:

- $(r_s, v_s, 0) \rightsquigarrow (r_f, v_f, 0)$;
- $(r_f, v_f, 0) \rightsquigarrow (r_f, v_f, 1)$,

ja võtma neist lühima.

Kuidas leida lühimat teekonda kahe tipu vahel suunatud graafis? Oletame, et tahame näiteks leida lühimat teekonda s ja f vahel. Teeme massiivi dist; paneme alguses $\text{dist}[s] = 0$ ja $\text{dist}[u] = \infty$ teiste u-de korral. Eesmärk on see, et lõpuks oleks iga u jaoks $\text{dist}[u]$ kaugus tipust s tippu u. Hoiame meeles järjekorda, kus on esialgu ainult tipp s. Teeme järgmist:

- võtame järjekorra esimese elemendi u ja eemaldame ta järjekorrast;
- iga serva $u \to v$ kohta:
  - kui $\text{dist}[v] = \infty$:
    - $\text{dist}[v] \leftarrow \text{dist}[u] + 1$;
    - lisame v järjekorra lõppu.

Lõpuks lühima teekonna pikkus on siis $\text{dist}[f]$. Kui on vaja teada ka teekonda $s \rightsquigarrow f$, siis hoiame iga tipu jaoks ühtlasi meeles, millise tipu kaudu temani jõuti. Siis saame tipust f "tagurpidi" tagasi tulla. Selle keerukus on $\mathcal{O}(E)$, kus E on graafi servade arv. Antud juhul on keerukus siis $\mathcal{O}(RC)$, sest meil on $2RC$ tippu, millest igaüks tekitab kuni 16 serva.

4. Suurtükid (`svejk`) 1 sek 50 punkti

Vaatleme kõiki N arvu permutatsioone, kus järjestikuste elementide vahede absoluutväärtuste summa on suurim võimalik. Leia:

1. see suurim võimalik summa;
2. selliste permutatsioonide arv;
3. sellistest permutatsioonidest leksikograafiliselt vähim;
4. sellistest permutatsioonidest leksikograafiliselt M-s.

($1 \leqslant N \leqslant 10^5$, $1 \leqslant M \leqslant 10^{18}$)

Lahendus

Uurime esiteks, missugused need maksimaalse summaga permutatsioonid on.

Lemma 1. *Kui P on optimaalne järjestus, siis pole ühtegi sellist i, et $P[i-1] < P[i] < P[i+1]$ ega $P[i-1] > P[i] > P[i+1]$.*

Tõestus. Võtame mingi järjestuse P. Valime vähima sellise i, et $P[i-1] < P[i] < P[i+1]$ või $P[i-1] > P[i] > P[i+1]$. Üldisust kitsendamata eeldame, et $P[i] < P[i+1]$. Vahetame omavahel ära $P[i]$ ja $P[i+1]$. Juhtub üks kolmest asjast:

- $P[i] > P[i+1] > P[i+2]$ ja vastus jääb samaks;
- $P[i] > P[i+1] < P[i+2]$ ja vastus kasvab.
- $i+1 = N$ ja vastus kasvab.

Igal juhul see "vähim i" suureneb vähemalt ühe võrra. Kordame seda protseduuri nii kaua, kuni ei ole enam selliseid i väärtusi. Vähemalt ühe korra vastus kasvas. Seega suvalisest järjestusest saab teha parema järjestuse, kus neid kasvavaid ega kahanevaid kolmikuid ei ole. □

See tähendab, et optimaalsed järjestused "sik-sakitavad": kui välja jätta $P[1]$ ja $P[N]$, siis igal elemendil on naabrid kas temast mõlemad suuremad või mõlemad väiksemad.

Iga paar $i, i+1$ liidab summale $|P[i] - P[i+1]|$. Kui $P[i] > P[i+1]$, siis see liidetav on $P[i] - P[i+1]$; kui $P[i] < P[i+1]$, siis see liidetav on $P[i+1] - P[i]$. Seega:

- kui $P[i]$ on mõlemast oma naabrist suurem, siis see liidab summasse $2P[i]$;
- kui $P[i]$ on mõlemast oma naabrist väikesem, siis see lahutab summast $2P[i]$.

Näeme, et on optimaalne panna "suured" arvud kohtadesse, kus nad on mõlemast oma naabrist suuremad, ning "väikesed" arvud kohtadesse, kus nad on mõlemast oma naabrist väikesemad. Otspunktidesse on optimaalne panna "keskmised" arvud.

Pärast mõningast mõtlemist jõuame järeldusele, et:

- Kui N on paaris, siis optimaalne permutatsioon on ühel järgmistest kujudest:

$$\begin{array}{rccccccl} C & \wedge & \vee & \cdots & \wedge & \vee & C+1 \\ C+1 & \vee & \wedge & \cdots & \vee & \wedge & C \end{array}$$

- Kui N on paaritu, siis optimaalne permutatsioon on ühel järgmistest kujudest:

$$\begin{array}{rcccccccl} C-1 & \wedge & \vee & \cdots & \wedge & \vee & \wedge & C \\ C & \vee & \wedge & \cdots & \vee & \wedge & \vee & C+1 \\ C & \wedge & \vee & \cdots & \wedge & \vee & \wedge & C-1 \\ C+1 & \vee & \wedge & \cdots & \vee & \wedge & \vee & C \end{array}$$

Siin $C = \left\lceil \frac{N}{2} \right\rceil$ ning $\wedge$ ja $\vee$ tuleb asendada vastavalt "suurte" ja "väikeste" arvudega. Siin on suured arvud need, mis on kummastki otspunktist suuremad ning väikesed arvud on need, mis kummastki otspunktist väiksemad. Seejuures on selge, et suurte arvude omavaheline järjekord ei ole oluline ning väikeste arvude omavaheline järjekord ka ei ole oluline.

Esimesed kolm ülesandeosa lahenduvad selle põhjal vahetult:

1. Optimaalse kauguste summa leidmiseks konstrueerime suvalise sellise permutatsiooni ja arvutame selle abil summa välja (või tuletame valemi).

2. Optimaalsete permutatsioonide arvu leidmiseks paneme tähele, et optimaalses permutatsioonis on $\lceil \frac{N-2}{2} \rceil$ "suurt" ja $\lfloor \frac{N-2}{2} \rfloor$ "väikest" arvu või vastupidi. Suuri arve omavahel võime suvaliselt järjestada, väikeseid ka. Otspunktide valikuks on vastavalt paarsusele 2 või 4 valikut, seega vastus on

$$\begin{cases} 2 \lceil \frac{N-2}{2} \rceil! \lfloor \frac{N-2}{2} \rfloor!, & \text{kui } N \text{ on paaris;} \\ 4 \lceil \frac{N-2}{2} \rceil! \lfloor \frac{N-2}{2} \rfloor!, & \text{kui } N \text{ on paaritu;} \end{cases}$$

3. Leksikograafiliselt vähima leidmiseks valime esimese otspunkti võimalikult väikese, sorteerime "suured" arvud kasvavalt ning "väikesed" arvud ka kasvavalt.

Aga kuidas leida leksikograafiliselt M-ndat? Oletame, et teame, missugused on leksikograafiliselt M-nda k esimest arvu ja on vaja arvutada arv $k+1$. Selle k esimese arvu valikuga oleme juba "mööda läinud" P permutatsioonist. Kui me paneme $k+1$-ndaks mingi arvu, siis võime lihtsasti välja arvutada, mitmest permutatsioonist me veel mööda läheme. Valime selle arvu piisavalt suure, et me ei lähe M-ndast permutatsioonist mööda, kuid läheme võimalikult paljudest permutatsioonidest mööda.

Keerukus $\mathcal{O}(N \log N)$ vms.

5. Koalitsioonilepped (`koal`) 1 sek 60 punkti

Loendada, mitu erinevat võimalikku koalitsiooni saavad moodustada K parteid parlamendis, kus on N saadikut. Parteid saavad moodustada koalitsiooni, kui neil on parlamendis häälteenamus, kusjuures selle häälteenamuse saavutamiseks on vaja kõiki koalitsiooni parteisid. ($1 \leqslant N \leqslant 10^{18}$, $1 \leqslant K \leqslant 36$)

Alamülesanne $N \leqslant 10^6$

Vaatleme mingit parteide rühma, millel on parlamendis häälteenamus. See saab moodustada koalitsiooni, kui sellest rühmast ühe partei välja arvamine võtab rühmalt häälteenamuse ära. Paneme tähele, et see rühm saab moodustada koalitsiooni täpselt siis, kui sellest rühmast kõige väikesema partei välja arvamine võtab rühmalt häälteenamuse ära.

Sorteerime parteid suurimast vähimani. Edaspidi mõtleme "partei 1" all suurimat, "partei 2" all suuruselt teist jne. Samamoodi M_1 on suurima partei liikmete arv, M_2 suuruselt teise liikmete arv jne.

Teeme $(K+1) \times (N+1)$ tabeli nimega dp. Meie eesmärk on seda tabelit täita nii, et $\mathrm{dp}[k][i]$ loendaks, mitu on selliseid parteide rühmi, kus:

- kõik parteid on parteide $1 \ldots k$ seast;
- rühmas on kokku i liiget.

Seejuures võime mõelda nii, et $\mathrm{dp}[0][0] = 1$ ja $\mathrm{dp}[0][i] = 0$ muude i-de korral.

Kuidas seda tabelit nii täita? Paneme tähele, et kui tabel on õigesti täidetud, siis peaks kehtima järgmine:

$$\mathrm{dp}[k][i] = \begin{cases} \mathrm{dp}[k-1][i], & \text{kui } i < M_k; \\ \mathrm{dp}[k-1][i] + \mathrm{dp}[k-1][i-M_k], & \text{muudel juhtudel.} \end{cases}$$

See on nii sellepärast, et on kaht tüüpi parteide rühmi, mis moodustuvad parteidest $1 \ldots k$ ja kus on i inimest.

- *Need, kuhu partei k ei kuulu.* Neid on $\mathrm{dp}[k-1][i]$ tükki;
- *Need, kuhu partei k kuulub.* Siis M_k inimest sealt on parteist k ja $i - M_k$ inimest parteidest $1 \ldots k-1$. Niisiis on neid $\mathrm{dp}[k-1][i-M_k]$ tükki.

Seda teades võime lihtsalt tabeli järjest täita keerukusega $\mathcal{O}(KN)$.

Kuidas siit nüüd aga ülesande vastus välja võluda? Märkame, et $\mathrm{dp}[k][i] - \mathrm{dp}[k-1][i]$ on selliste rühmade arv, kus k on kõige väiksem partei ja kus on i inimest. Selline rühm saab moodustada koalitsiooni parajasti siis, kui i on häälteenamus, aga $i - M_k$ ei ole. Seega: käime kõik k, i uuesti läbi. Kui i on häälteenamus, aga $i - M_k$ mitte, siis liidame vastusele $\mathrm{dp}[k][i] - \mathrm{dp}[k-1][i]$. Keerukus $\mathcal{O}(KN)$.

Täislahendus

Nagu eelmises osas, sorteerime parteid suuremast väiksemani. Jaotame parteid kaheks: $\frac{K}{2}$ "suureks parteiks" ja $\frac{K}{2}$ "väikeseks parteiks".

Loendame esmalt kõik koalitsioonid, mis koosnevad ainult suurtest parteidest. Seda saame teha keerukusega $\mathcal{O}(K2^{\frac{K}{2}})$, kui käime lihtsalt läbi kõikvõimalikud erinevad suurtest parteidest koosnevad rühmad, ja kontrollime igas rühmas, kas see on koalitsiooni tingimustele vastav. Samamoodi saame üle lugeda koalitsioonid, mis koosnevad ainult väikestest parteidest.

Ülejäänud võimalikes koalitsioonides on nii suuri kui väikeseid parteisid. Olgu meil üks väikeste parteide hulga mittetühi alamhulk A. Tahame üle lugeda, kui palju on selliseid võimalikke suurte parteide hulki B, et A parteid koos B parteidega saavad koalitsiooni moodustada. Olgu s_A hulga A parteide saadikute arv, s_B hulga B parteide saadikute arv ning s_l hulga A väikseima partei saadikute arv. Varasemalt veedusime, et koalitsiooni saab moodustada siis, kui $s_A + s_B > \frac{N}{2}$ ja $s_A + s_B - s_l \leqslant \frac{N}{2}$. Tähendab, koalitsiooni saab moodustada siis, kui

$$\frac{N}{2} - s_A < s_B \leqslant \frac{N}{2} - s_A + s_l.$$

Niisiis peame iga A jaoks loendama, mitu suurte parteide hulka B on sellised, et s_B jääb intervalli $\left(\frac{N}{2} - s_A, \frac{N}{2} - s_A + s_l\right]$.

Algoritm on praegu selline:

1. Käime läbi kõik võimalikud suurte parteide rühmad B:
   (a) Kui rühm B saab koalitsiooni moodustada, siis lisame vastusele 1;
   (b) Paneme s_B kirja massiivi s;
2. Käime läbi kõik võimalikud väikeste parteide rühmad A:
   (a) Kui rühm A saab koalitsiooni moodustada, siis lisame vastusele 1;
   (b) Arvutame rühma A vähima partei liikmete arvu, tähistame s_l.
   (c) Loeme, mitu elementi s_B massiivis s on sellised, et
   $$s_B \in \left(\frac{N}{2} - s_A, \frac{N}{2} - s_A + s_l\right].$$
   Liidame nende arvu vastusele.

Kuidas leida, mitu suurte parteide hulka jääb mingisse vastavasse intervalli (samm 2c). Unustades detailid ära, saame me järgmise ülesande: On antud $M = 2^{\frac{K}{2}} - 1$ arvu $s_1, s_2, s_3, \ldots, s_M$ ja Q

päringut kujul "antud arvud l ja r; leida, mitu arvude $s_1, s_2, s_3, \ldots, s_M$ seast jäävad intervalli $(l, r]$." Veel lihtsamalt, me võime mõelda, et meil on hoopis päringud "antud arv x; leida, mitu arvude $s_1, s_2, s_3, \ldots, s_M$ on $\leqslant x$." Seda saab teha nn kahendotsingu meetodil. Sorteerime arvud $s_1, s_2, \ldots, s_M$ kasvavalt. Nüüd võime hoopis vastata päringutele "antud arv x, leia viimane arvude $s_1, s_2, \ldots, s_M$ seast, mis on $\leqslant x$."

Vaatleme massiivi s keskmist elementi. Kui see on suurem, kui x, jääb otsitav indeks vasakule poole keskmist elementi, vastasel juhul paremale. Saame poole massiivist "ära visata". Me võime seda tegevust korrata, kuni vaatluse all on ainult üks element. Tegime $\mathcal{O}(\log M) = \mathcal{O}(\log 2^{\frac{K}{2}}) = \mathcal{O}(K)$ poolitust.

Kokku keerukus $\mathcal{O}(K 2^{\frac{K}{2}})$.

6. Tordi lõikamine (`tort`)

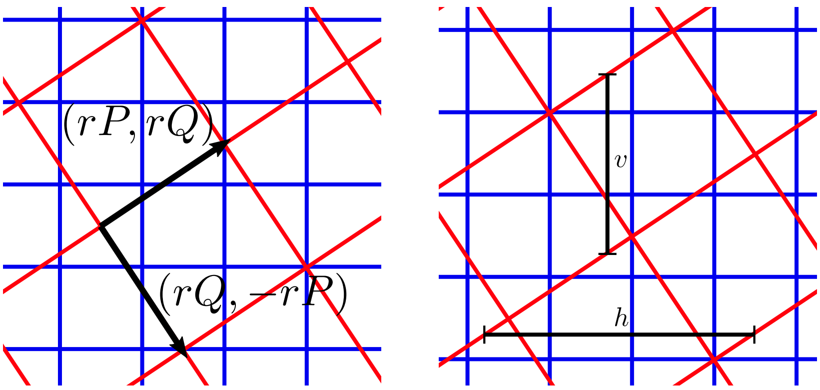
1 sek / 10 sek 60 punkti

On antud $M \times N$ ühikruudustik. Selle peale on asetatud nurga $\arctan\left(\frac{P}{Q}\right)$ all teine ruudustik, kus iga ruudu pindala on S. Seejuures teame, et ühikruudustiku all vasakus nurgas ristuvad ka teise ruudustiku jooned. Mitu kujundit kokku tekib? ($1 \leqslant M, N, S \leqslant 10^9$, $1 \leqslant P, Q \leqslant 10^3$, $\sqrt{\frac{S}{P^2+Q^2}} \in \mathbb{Q}$)

Lahendus

Edaspidi kutsume ühikruudustikku *siniseks ruudustikuks* ja seda teist *punaseks ruudustikuks*. Kõik koordinaadid on antud sinise ruudustiku järgi.

Esmalt tuletame valemid, mida hiljem palju kasutame.



(a) (b)

Joonis 1

Lemma 2. *Tähistame $r = \sqrt{\frac{S}{P^2+Q^2}}$. Ülesande tingimus väidab, et see on ratsionaalarv.*

1. *Liikudes mööda punast joont ühe sammu "paremale üles" oleme liikunud mööda x-telge rP ja mööda y-telge rQ;*
2. *Liikudes mööda punast joont ühe sammu "paremale alla" oleme liikunud mööda x-telge rQ ja mööda y-telge $-rP$.*

Tõestus. Punase ruudu küljepikkus on $\sqrt{S}$.

1. Joonistame täisnurkse kolmnurga, mille hüpotenuus on punase ruudu küljepikkus ja kaatetid on paralleelsed vastavalt x- ja y-teljega. Olgu kaatetite pikkused a ja b. Nurgast punaste ja siniste joonte vahel teame, et $\frac{a}{b} = \frac{P}{Q}$ ehk $a = \frac{bP}{Q}$. Pythagorase teoreemist
$$S = a^2 + b^2 = \frac{b^2P^2}{Q^2} + b^2 = b^2\frac{P^2+Q^2}{Q^2}$$
$$b = Q\sqrt{\frac{S}{P^2+Q^2}} = rQ,$$
ja seega $a = \frac{bP}{Q} = \frac{rQP}{Q} = rP$.
2. Analoogiline.

□

Lemma 3. *Oletame, et "paremale üles" suunatud punane sirge, läbib punkti (x, y). Siis x-telje suunas "järgmine" temaga paralleelne sirge läbib punkti $(x + h, y)$, kus $h = r\frac{P^2+Q^2}{Q}$.*

Analoogiliselt, y-telje suunas "järgmine" läbib punkti $(x, y + v)$, kus $v = r\frac{P^2+Q^2}{P}$.

Tõestus. Geomeetriliselt on selge, et (x, y) väärtusest h ei sõltu. Liigutame mööda punast joont (x, y) punkti, kus punased jooned lõikuvad.

Liigume sealt mööda punast joont ühe sammu paremale alla. Jõuame punkti $(x + rQ, y - rP)$. Nüüd peame liikuma mööda punast joont paremale üles nii kaua, kuni meie y-koordinaat on jälle y. Selleks peame tegema $\frac{rP}{rQ} = \frac{P}{Q}$ sammu paremale üles. Seejuures x-koordinaat nihkub $\frac{P}{Q} \cdot rP = \frac{rP^2}{Q}$ võrra. Oleme jõudnud punkti $(x + rQ + \frac{rP^2}{Q}, y) = (x + r\frac{P^2+Q^2}{Q}, y) = (x + h, y)$.

Vertikaalne juht on analoogiline. □

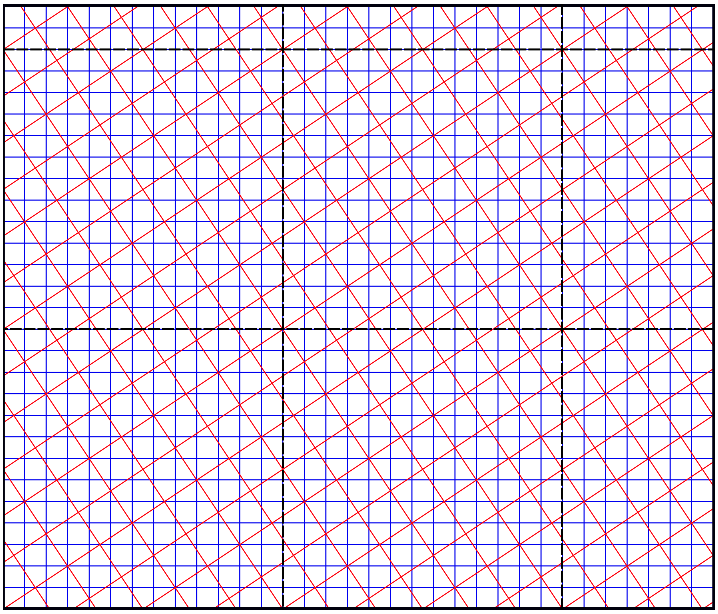
Alustame ülesande lahendamist sellest, et teeme ülesande väikesemaks. Meenutame, et punased jooned ristuvad punktis $(0, 0)$, ja sinised jooned ristuvad samuti selles punktis. Kui liigume paremale mööda x-telge ja mingis punktis uuesti punased jooned ristuvad ja sinised jooned ristuvad samas punktis, siis hakkab uuesti sama muster peale (vt. joonis 2).

Lemma 4. *Punased jooned ristuvad punktis $(K, 0)$, kus $K = \sqrt{S}\sqrt{P^2 + Q^2} \in \mathbb{Z}$.*

Tõestus. Alustame ruudust $(0, 0)$. Tahame teha mööda punast joont i sammu paremale üles ning j sammu paremale alla nii, et langeme x-teljele ja täisarvuliste koordinaatidega ruudule. Niisiis otsime täisarve i, j nii, et
$$irP + jrQ \in \mathbb{Z};$$
$$irQ - jrP = 0.$$

Teisest tingimusest saame $irQ = jrP$ ehk $iQ = jP$. Avaldame $j = \frac{iQ}{P}$. Saame, et peab kehtima $irP + \frac{iQ^2r}{P} = ir\left(\frac{P^2+Q^2}{P}\right) \in \mathbb{Z}$. Ülesande tingimus ütleb, et r^2 esitub kujul $\frac{p^2k}{q^2k}$, kus p, q, k on täisarvud, $p^2k = S$, $q^2k = P^2 + Q^2$. Seega
$$ir\left(\frac{P^2+Q^2}{P}\right) = i\frac{p}{q}\frac{q^2k}{P} = i\frac{pqk}{P}.$$

Valides $i = P$, on see täisarv. Ka j peab täisarv olema, kontrollime: $j = \frac{iQ}{P} = Q \in \mathbb{Z}$.

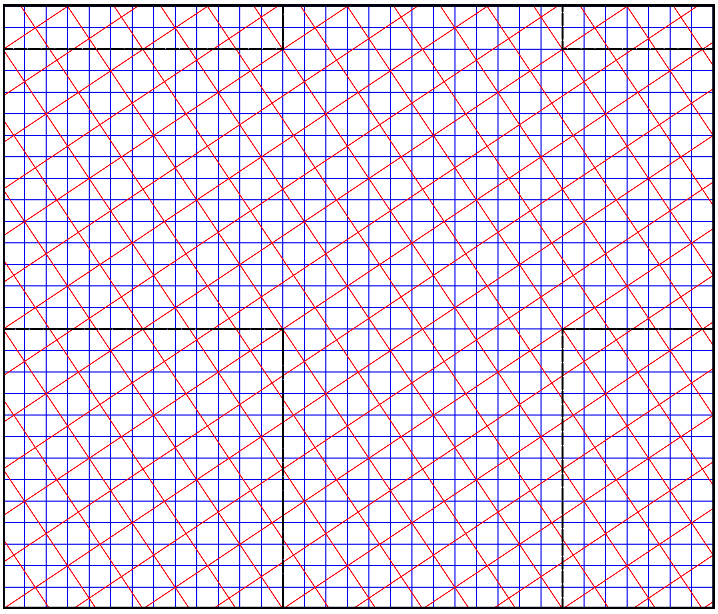


Joonis 2

Oleme konstrueerinud oma punkti $K = pqk$. Seejuures

$$K = pqk = pq\sqrt{k}\sqrt{k} = \sqrt{p^2k}\sqrt{q^2k} = \sqrt{S}\sqrt{P^2 + Q^2}.$$

□

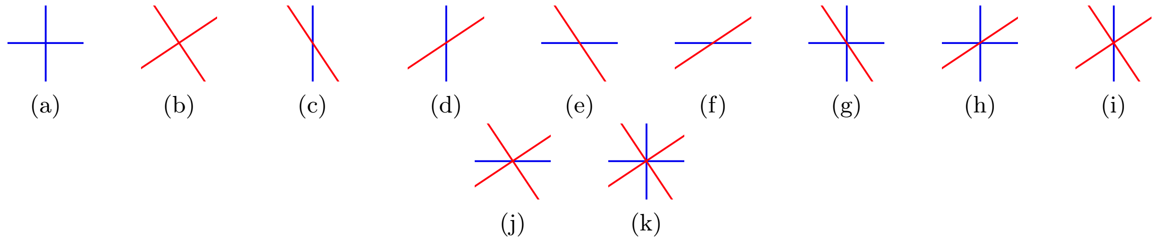


Joonis 3

See tähendab, et võime lahendada ülesande nende ristkülikute jaoks, mis on joonisel 3 musta raamiga ümbritsetud. Sisuliselt vähendasime ülesande piire. Edaspidi eeldame lihtsalt hoopis, et $1 \leqslant M, N \leqslant \sqrt{S}\sqrt{P^2 + Q^2} \leqslant \sqrt{10^9}\sqrt{10^3 + 10^3} \approx 4.5 \cdot 10^7$.

Kujutame ette, et meil on sinine ruudustik ja oleme sinna juba lisanud mõned punased diagonaalid. Lisame ühe uue punase diagonaali. Vaatame, mitu erinevat lõikepunkti moodustab see diagonaal olemasolevate sirgetega. Kui diagonaal lõikab ühes punktis mitut sirget, siis loeme seda ikkagi üheks lõikepunktiks. Tähistame selle lõikepunktide arvu c. Paneme tähele, et selle diagonaali lisamisel kasvas ruudustikus olevate piirkondade arv $c-1$ võrra.

Seega peame tegelikult vaid loendama, mitu lõikepunkti lõpptulemuses on; seejuures pidades meeles, mitu sirget iga lõikepunkti läbib. Lõikepunkte endid on 11 erinevat "tüüpi", samas enamus nendest tüüpidest on väga sarnased.



Joonis 4: Lõikepunktide tüübid

Kirjeldame esmalt, kuidas loendada tüüpi (i) olevate lõikude arvu. Tüübid (c) kuni (k) lahenduvad sellega täiesti analoogiliselt. Käime selleks läbi $\mathcal{O}(M)$ vertikaalset sirget ning iga sirge kohta loendame eraldi, mitu lõikepunkti sellel sirgel on. Võtame ühe vertikaalse sirge $x = x_0$. Paremale üles suunatud punased jooned läbivad seda punktides $(x_0, y_1), (x_0, y_1 + v), (x_0, y_1 + 2v), \ldots$ ning vasakule üles suunatud jooned punktides $(x_0, y_2), (x_0, y_2 + h), (x_0, y_2 + 2h), \ldots$. Siin y_1 ja y_2 on ratsionaalarvud, mida on kerge välja arvutada. Tahame teada, kui paljud nendest lõikepunktidest langevad kokku. Olgu p arvude y_1, y_2, v, h nimetajate vähim ühiskordne. Peame loendama, mitu lahendit on kongruentside süsteemil

$$\begin{cases} y & \equiv py_1 \pmod{pv}; \\ y & \equiv py_2 \pmod{ph}, \end{cases}$$

kus $0 \leqslant y \leqslant pN$. Meenutame Hiina jäägiteoreemi:

Teoreem. *Kongruentside süsteem*

$$\begin{cases} y & \equiv a \pmod{m}; \\ y & \equiv b \pmod{n} \end{cases}$$

ei lahendu, kui $a \not\equiv b \pmod{\gcd(m, n)}$. *Vastasel juhul lahendub ta üheselt mooduli* $\operatorname{lcm}(m, n)$ *järgi; lahendiks on*

$$y = \frac{avn + bum}{\gcd(m, n)}.$$

Siin u, v *on sellised täisarvud, et* $um + vn = \gcd(m, n)$. *Neid on võimalik leida nn laiendatud Eukleidese algoritmiga, keerukus* $\mathcal{O}(\log(m + n))$.

Nii leiame perioodi ja esimese lõikepunkti. Edasi on kerge leida tüüpi (i) lõikepunktide arvu sirgel $x = x_0$. Mõned nii leitud tüüpi (i) lõikepunktid on tegelikult hoopis tüüpi (k), seetõttu tuleb pärast tüüpi (i) lõikepunktide arvu leidmist lahutada maha tüüpi (k) lõikepunktide arv.

Veel tuleb leida tüüpi (b) lõikepunktide arv. Elegantne viis seda teha on kasutada nn Picki teoreemi:

Teoreem. *Olgu (ennast mitte lõikava) hulknurga pindala* A*, piirjoonel olevate täisarvuliste koordinaatidega punktide arv* b *ja sisemuses olevate täisarvuliste koordinaatidega punktide arv* i*. Siis*

$$A = i + \frac{b}{2} - 1.$$

Seekord kasutame sellist koordinaadisüsteemi, kus koordinaatteljed on paralleelsed punaste sirgetega.

Keerukus $\mathcal{O}(\sqrt{S}\sqrt{P^2+Q^2}\log(P^2+Q^2))$; ettevaatliku implementatsiooni korral saab isegi teha $\mathcal{O}(\sqrt{S}\sqrt{P^2+Q^2})$. Mõlemad lähevad läbi.

7. Kokkuhoidlik testimine (`test`) 60 punkti

Antud on ülesanne: pane 8 numbrist kokku kaks legaalset kellaaega nii, et nende aegade summa oleks maksimaalne.

On antud 100 vigast lahendust. Konstrueeri võimalikult väike arv teste, millega saab kõik 100 vigast lahendust läbi kukutada.

Valelahenduste struktuurist

Kui valede lahenduste koodile natuke peale vaadata, siis märkame, et need on peaaegu ühesugused. Nad erinevad ainult üle-eelviimase rea poolest, mis on igas lahenduses erinev 112 bitist koosnev "mask".

Mis on selle maski roll? Kõigepealt selgitame üht korrektset lahendust.

- Käime läbi kõik C_8^4 viisi jaotada kaardid kaheks 4-kaardiliseks hulgaks:
  - Kummaski hulgas käime läbi kõik 4! viisi neid järjekorda panna ja valime suurima ajaga legaalse.

Aga võime tähele panna, et mõned 4! permutatsioonist ei ole kunagi optimaalsed.

Lemma 5. *Oletame, et meil on antud arvud d_0, d_1, d_2, d_3. Sorteerime nad nii, et $d_0 \leqslant d_1 \leqslant d_2 \leqslant d_3$. Alati on optimaalne üks järgmistest permutatsioonidest:*

$$\begin{array}{cccc}
(d_3,d_2,d_1,d_0), & (d_2,d_3,d_1,d_0), & (d_2,d_1,d_3,d_0), & (d_2,d_1,d_0,d_3), \\
(d_1,d_3,d_2,d_0), & (d_1,d_3,d_0,d_2), & (d_1,d_2,d_3,d_0), & (d_1,d_2,d_0,d_3), \\
(d_1,d_0,d_3,d_2), & (d_1,d_0,d_2,d_3), & (d_0,d_3,d_2,d_1), & (d_0,d_3,d_1,d_2), \\
 & (d_0,d_1,d_3,d_2), & (d_0,d_1,d_2,d_3). &
\end{array}$$

Tõestus. (Automaatne) juhtude läbivaatus. □

Näiteks, kui on optimaalne paigutada d_3 esimeseks, siis $d_3 \leqslant 2$ ja on selge, et on optimaalne paigutada numbrid kahanevas järjekorras, sest siis kõik arvud on ülimalt 2 ja "mahuvad" igale poole. Antud valed lahendused on ehitatud korrektse lahenduse peale, mis vaatab ainult neid 14 permutatsiooni.

Maskis:

- 70 bitti näitavad, millised $C_8^4 = 70$ kombinatsioonist läbi vaadatakse;
- 14 bitti näitavad, millised 14 permutatsioonist esimese kella jaoks läbi vaadatakse (iga permutatsiooni jaoks on oma bitt);
- 14 bitti näitavad, millised 14 permutatsioonist teise kella jaoks läbi vaadatakse;
- ülejäänud 14 bitti ei näita midagi.

Korrektse lahenduse saame siis, kui paneme maskis kõik bitid ühtedeks.

Suund lahenduseks

Väga üldiselt rääkides võiks lahendus olla midagi järgmise skeemi moodi:

1. Genereeri suur arv teste.
2. Jooksuta kõiki valelahendusi nende testide peal ja vaata, millised valelahendused milliste testide peal kukuvad.
    - Kui kaks testi kukutavad täpselt samad lahendused läbi, võime ühe nendest kohe ära visata.
3. Nüüd on meil palju teste, mis "katavad" erinevaid lahendusi. Pane nendest kokku komplekt, mis katab kõik lahendused.

Võib olla meelitav mõte proovida genereerida kõik 10^8 erinevat testi ja panna programmid nende peal jooksma. Kahjuks võib see liiga kaua aega võtta.

Võiks proovida genereeritavate testide arvu vähendada ja testimise protsessi kiirendada, kasutades seda, mis on valede lahenduste sees. Näiteks, kuna valed lahendused jaotavad numbrid kaheks osaks ja siis sorteerivad kummagi osa, tähendab see, et mõned testid on samaväärsed. Me võime kasutada ainult neid sisendteste, kus esimesed neli numbrit on sorteeritud.

Teiseks, meil ei ole mõtet 100 erinevat vale lahendust iga testi peal jooksutada. Mõistlikum on võtta test ja jooksutada seda õige lahenduse peal, pidades meeles, missugused kombinatsioonide ja permutatsioonide kolmikud saavutavad optimaalse tulemuse. Nii saame me kiiresti kontrollida, kas test kukutab antud vale lahenduse läbi või mitte.

Alternatiivina võib proovida ka lihtsalt juhuslikke teste genereerida. Samas nii ei pruugi optimaalset tulemust saada. Kui vaadata läbi näiteks juhuslikult valitud $\frac{1}{5}$ kõikidest testidest, siis suure tõenäosusega ei saa vaadatud testidest optimaalset testikomplekti kokku panna.

Kui me vaatame (kuidagi) läbi kõik võimalikud testid, ja viskame välja sellised testid, mis kukutavad täpselt samad lahendused läbi, jääb alles 679 testi. Mõni test on mõnest teisest testist "rangelt parem": kukutab läbi samad lahendused ja enamgi veel. Kui viskame sellised "halvemad" testid ka välja, jääb alles 133 testi.

Nüüd on meil 100-elemendiline "universum" ja 133 selle alamhulka. Missugune on kõige väikesem alamhulkade pere, mis universumi katab? See on informaatikas klassikaline NP-täielik ülesanne, nn *set cover problem*. Täpset kiiret lahendust sellele ei ole, samas on palju uuritud, kuidas kõige paremini ligikaudset vastust leida.

Minimaalne vastus on 18 testi. Uurige, katsetage!