1. Alamjada (algajate rühm)

Kõige ilmsem võimalus selle ülesande lahendamiseks on vaadelda järjest jada kõiki elemente a_i ja loendada iga a_i jaoks, kui palju elemente võime võtta alamjadasse $a_i, a_{i+1}, \ldots,$ et see oleks ikka veel kasvav. Selline, failis `ajadalah0.pas` toodud lahendus on küll õige, kuid mitte eriti efektiivne. Kui kogu esialgne jada on kasvav, loendame elemendist a_1 alustades kasvava alamjadana N elementi, elemendist a_2 alustades $N-1$ elementi j.n.e, kulutades kõigi N alamjada uurimiseks kokku umbes N^2 sammu.

Parema lahenduse saame, pannes tähele, et kui mingi a_i kohta on teada, et maksimaalne sellest elemendist algav kasvav alamjada on k-elemendiline, siis tähendab see, et $a_{i+k-1} \geqslant a_{i+k}$ ja iga $i < j < i + k$ korral peab elemendist a_j algav kasvav alamjada samuti lõppema elemendiga a_{i+k-1} ja seega olema lühem kui elemendist a_i algav alamjada. Seega peame maksimaalse kasvava alamjada leidmiseks vaatlema hoopis mittekasvavaid paare $a_j \geqslant a_{j+1}$ — iga selline paar lõpetab ühe kasvava alamjada. Selliste mittekasvavate paaride leidmiseks vaatleme iga elementi ühe korra ja kulutame selleks kokku umbes N sammu.

Selle idee realiseerimisel tuleb olla tähelepanelik, et mitte jätta vaatlemata antud jada viimast elementi — sellele järgnevat ju pole, seega pole teda ka millegagi võrrelda. Üks võimalus on pärast jada läbivaatust eraldi kontrollida viimase elemendiga lõppeva alamjada pikkust, nagu on tehtud failis `ajadalah1.pas` toodud lahenduses. Teine võimalus on tekitada enne töö algust jada lõppu fiktiivne mittekasvav paar — näiteks lisades sinna esialgse jada viimase elemendi koopia. Selline lahendus on toodud failis `ajadalah2.pas`.

Testid

1. $N = 1$. Vastus $K = 1$, $a_{1\ldots1}$. Minimaalne test. 3 punkti.
2. $N = 5$. Vastus $K = 3$, $a_{1\ldots3}$. Üks maksimaalse pikkusega kasvav alamjada, antud jada alguses. 3 punkti.
3. $N = 5$. Vastus $K = 3$, $a_{2\ldots4}$. Üks maksimaalse pikkusega kasvav alamjada, antud jada keskel. 3 punkti.
4. $N = 5$. Vastus $K = 3$, $a_{3\ldots5}$. Üks maksimaalse pikkusega kasvav alamjada, antud jada lõpus. 3 punkti.
5. $N = 9$. Vastus $K = 3$, $a_{6\ldots8}$. Mitu maksimaalse pikkusega kasvavat alamjada. 3 punkti.
6. $N = 9$. Vastus $K = 9$, $a_{1\ldots9}$. Piirjuht: antud jada on kasvav. 3 punkti.
7. $N = 9$. Vastus $K = 1$, $a_{9\ldots9}$. Piirjuht: antud jada on kahanev. 3 punkti.
8. $N = 9$. Vastus $K = 4$, $a_{6\ldots9}$. Jadas on kõrvuti võrdsed elemendid. Need ei ole kasvavad. 3 punkti.
9. $N = 100$. Vastus $K = 4$, $a_{78\ldots81}$. Keskmise suurusega juhuslik test. 3 punkti.
10. $N = 10\,000$. Vastus $K = 6$, $a_{426\ldots431}$. Maksimaalne juhuslik test. 3 punkti.

Kokku 30 punkti.

2. Alamjada (edasijõudnute rühm)

Failis `ejadalah1.pas` toodud lahendus on sisuliselt jada sorteerimine pistemeetodil, millele on lisatud väike täiendus aritmeetilisuse kontrollimiseks. Pistemeetod on klassikaline sorteerimisalgoritm, mis vaatleb antud jada esimest elementi kui 1-elemendilist sorteeritud alamjada ja hakkab seda kasvatama tagapool olevate elementide sobivatesse kohtadesse vahelepistmisega. Iga elemendi lisamise järel kontrollime, kas sorteeritud alamjada on aritmeetiline jada või mitte. Sorteerimise lõpuks on meil teada pikima aritmeetilise jada pikkus. Kasutades elementidega koos hoitavaid esialgseid järjekorranumbreid nopime nüüd juba järjestatud elementide hulgast välja need, mis olid esialgse jada K esimest. Kuna kogu jada on sorteeritud, on seda ka need K elementi. Seega võime nad kohe leidmise järjekorras faili kirjutada.

Testid

1. $N = 1$. Vastus $K = 1$. Minimaalne test. 3 punkti.
2. $N = 2$. Vastus $K = 2$. Peaaegu minimaalne test. 3 punkti.
3. $N = 3$. Vastus $K = 3$. Kogu jada on aritmeetiline. 3 punkti.
4. $N = 5$. Vastus $K = 3$. Väike lihtne test. 3 punkti.
5. $N = 5$. Vastus $K = 5$. Kogu jada on aritmeetiline. 3 punkti.
6. $N = 8$. Vastus $K = 5$. Väike lihtne test. 3 punkti.
7. $N = 15$. Vastus $K = 5$. Väike juhuslik test. 3 punkti.
8. $N = 100$. Vastus $K = 50$. Keskmise suurusega juhuslik test. 3 punkti.
9. $N = 300$. Vastus $K = 100$. Keskmise suurusega juhuslik test. 3 punkti.
10. $N = 1000$. Vastus $K = 300$. Maksimaalne juhuslik test. 3 punkti.

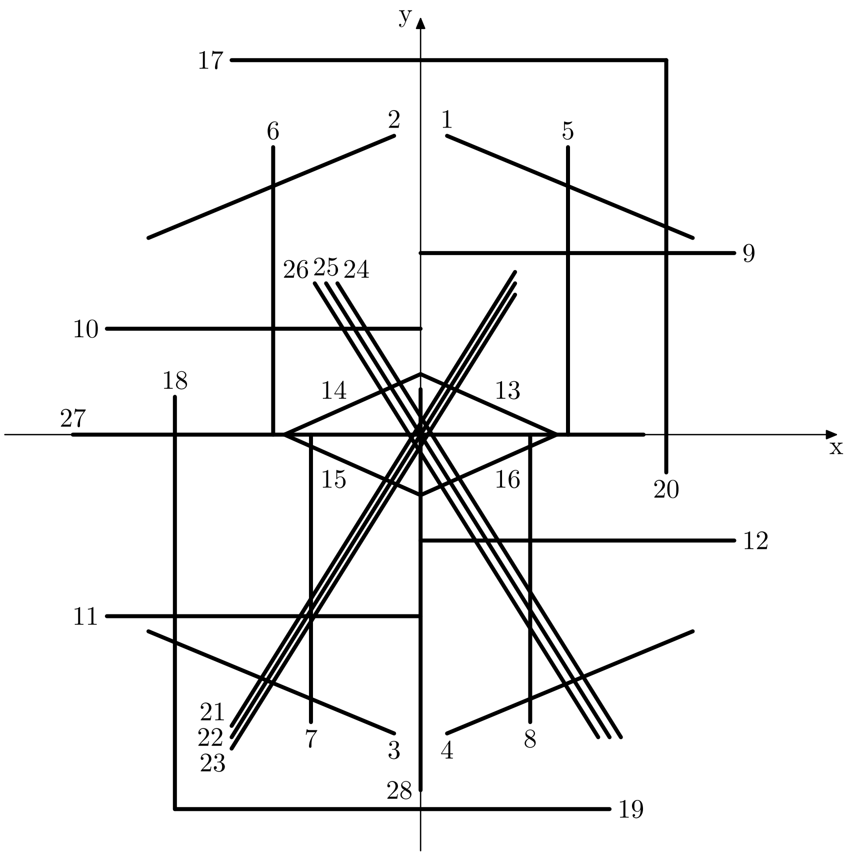
Kokku 30 punkti.

3. Sirglõik (algajate rühm)

Üks võimalus on lahendada see ülesanne "jaga ja valitse" meetodil — kui ülesanne on ühekorraga lahendamiseks liiga keeruline, siis jagame selle väiksemateks alamülesanneteks ja lahendame need ühekaupa. Käesoleval juhul on jagamine väga otsene — kui uuritava lõigu AB otspunktid on teine teisel pool y-telge, siis leiame selle lõigu ja y-telje lõikepunkti C ning edasi uurime lõike AC ja CB juba eraldi. Analoogiliselt tükeldame uuritava lõigu ka x-telje abil ja saame lõpuks jupid, mis on igaüks tervenisti ühe veerandi sees. Nende kuuluvust on juba lihtne kindlaks teha. Sellel ideel põhinev lahendus on toodud failis `aloiklah1.pas`.

Testid

Testid 1 kuni 28 graafiliselt (testide 29 ja 30 lõigud sellele joonisele ei mahu):



1. Kaldus lõik täielikult I veerandis. 1 punkt.
2. Kaldus lõik täielikult II veerandis. 1 punkt.
3. Kaldus lõik täielikult III veerandis. 1 punkt.
4. Kaldus lõik täielikult IV veerandis. 1 punkt.
5. Vertikaalne lõik täielikult I veerandis, üks otspunkt x-teljel. 1 punkt.
6. Vertikaalne lõik täielikult II veerandis, üks otspunkt x-teljel. 1 punkt.
7. Vertikaalne lõik täielikult III veerandis, üks otspunkt x-teljel. 1 punkt.
8. Vertikaalne lõik täielikult IV veerandis, üks otspunkt x-teljel. 1 punkt.
9. Horisontaalne lõik täielikult I veerandis, üks otspunkt y-teljel. 1 punkt.

10. Horisontaalne lõik täielikult II veerandis, üks otspunkt y-teljel. 1 punkt.
11. Horisontaalne lõik täielikult III veerandis, üks otspunkt y-teljel. 1 punkt.
12. Horisontaalne lõik täielikult IV veerandis, üks otspunkt y-teljel. 1 punkt.
13. Kaldus lõik täielikult I veerandis, üks otspunkt x-, teine y-teljel. 1 punkt.
14. Kaldus lõik täielikult II veerandis, üks otspunkt x-, teine y-teljel. 1 punkt.
15. Kaldus lõik täielikult III veerandis, üks otspunkt x-, teine y-teljel. 1 punkt.
16. Kaldus lõik täielikult IV veerandis, üks otspunkt x-, teine y-teljel. 1 punkt.
17. Horisontaalne lõik I ja II veerandi vahel. 1 punkt.
18. Vertikaalne lõik II ja III veerandi vahel. 1 punkt.
19. Horisontaalne lõik III ja IV veerandi vahel. 1 punkt.
20. Vertikaalne lõik IV ja I veerandi vahel. 1 punkt.
21. Kaldus lõik I ja III veerandi vahel, läbib II veerandit. 1 punkt.
22. Kaldus lõik I ja III veerandi vahel, läbib 0-punkti. 1 punkt.
23. Kaldus lõik I ja III veerandi vahel, läbib IV veerandit. 1 punkt.
24. Kaldus lõik II ja IV veerandi vahel, läbib I veerandit. 1 punkt.
25. Kaldus lõik II ja IV veerandi vahel, läbib 0-punkti. 1 punkt.
26. Kaldus lõik II ja IV veerandi vahel, läbib III veerandit. 1 punkt.
27. Horisontaalne lõik x-teljel. 1 punkt.
28. Vertikaalne lõik y-teljel. 1 punkt.
29. Kaldus lõik I ja III veerandi vahel, läbib II veerandit. Suured arvud. 1 punkt.
30. Kaldus lõik II ja IV veerandi vahel, läbib 0-punkti. Suured arvud. 1 punkt.

Kokku 30 punkti.

4. Sirglõik (edasijõudnute rühm)

Selle ülesande lahendamiseks võiks kõigepealt kindlaks teha, kas sirge, millel antud lõik asub, üldse ringjoonega lõikub. Kui ei lõiku, siis ei saa ka lõigu mingi osa ringi sees olla. Kui lõikub, siis tuleb kontrollida, kas kumbki lõigu otspunktidest on väljaspool ringjoont. Kui on, siis tuleb see otspunkt asendada lõikepunktiga, mis on talle lähemal. Sellega me lõikame lõigu otsast maha selle osa, mis jääb ringist välja. Kui lõik on niiviisi "ära pügatud", siis allesjääva osa pikkus ongi otsitav vastus. Selline lahendus on toodud failides `eloiklah1.pas` ja `eloiklah2.pas`. Nende kahe lahenduse erinevus on vaid lõigu esituses.

Mõlemad lahendused lähtuvad ringjoone definitsioonist, mille kohaselt ringjoone moodustavad parajasti need punktid, mille kaugus ringjoone keskpunktist on võrdne ringjoone raadiusega, ehk punktid, mille koordinaadid (x, y) rahuldavad võrrandit

$$(x - x_0)^2 + (y - y_0)^2 = r^2.$$

Failis `eloiklah1.pas` toodud lahendus kasutab sirge esitamiseks koolimatemaatikast tuttavat võrrandit $y = ux + v$. Sirge tõusu u võime avaldada antud lõigu otspunktide koordinaatidest: $u = (y_b - y_a)/(x_b - x_a)$ ja algordinaadi v valime sellise, et sirge läbiks punkti A: $v = y_a - ux_a$.

Sirge ja ringjoone lõikepunktid asuvad mõlemal joonel samaaegselt, ja peavad seega rahuldama ka mõlema joone võrrandeid. Asendades ringjoone võrrandisse y avaldise sirge võrrandist, saame lõikepunktide x-koordinaatide avaldamiseks ruutvõrrandi

$$(x - x_0)^2 + (ux + v - y_0)^2 = r^2,$$

mille võime sarnaseid liikmeid koondades viia lahendamiseks sobivamale kujule

$$[u^2 + 1]x^2 - 2[x_0 - u(v - y_0)]x + [x_0^2 + (v - y_0)^2 - r^2] = 0.$$

Selle lahenduse puudus on vajadus eraldi käsitleda vertikaalseid sirgeid, mida ei saa avaldada kujul $y = ux + v$. Failis `eloiklah1.pas` toodud lahendus kasutab ära asjaolu, et meid huvitab ainult ringi sisse jääva lõiguosa pikkus, ja lihtsalt vahetab x- ja y-teljed, muutes sellega vertikaalse sirge horisontaalseks. Muidugi võiks vertikaalsete sirgete juhu ka eraldi lahendada.

Failis `eloiklah2.pas` toodud lahendus seevastu lähtub tähelepanekust, et sirgel AB asuvad parasjagu need punktid, mille koordinaadid (x, y) avalduvad kujul

$$\begin{cases} x = (1 - t)x_a + tx_b, \\ y = (1 - t)y_a + ty_b. \end{cases}$$

(Mõnikord võib olla kasu ka teadmisest, et lõigul AB asuvad punktid rahuldavad lisaks tingimust $0 \leqslant t \leqslant 1$ ja t väärtused kasvavad punkti A poolt punkti B poole liikudes.) Asendades need koordinaatide avaldised eelmise juhuga sarnaselt ringjoone võrrandisse, saame sirge ja ringjoone lõikepunktidele vastavate t väärtuste avaldamiseks ruutvõrrandi

$$[(1 - t)x_a + tx_b - x_0]^2 + [(1 - t)y_a + ty_b - y_0]^2 = r^2,$$

mille võime sarnaseid liikmeid koondades viia lahendamiseks sobivamale kujule

$$[(x_b - x_a)^2 + (y_b - y_a)^2]t^2 + 2[(x_b - x_a)(x_a - x_0) + (y_b - y_a)(y_a - y_0)]t + [(x_a - x_0)^2 + (y_a - y_0)^2 - r^2] = 0.$$

5. Lennureis (algajate rühm)

Kõige lihtsam viis soodsaima reisi leidmiseks on vaadata läbi kõik lendudepaarid, kus üks lend on liinilt $A - B$ ja teine liinilt $B - C$. Selleks, et aegade võrdlemine selle läbivaatuse käigus lihtsam oleks, teisendame enne kõik ajad päeva algusest möödunud minutite arvudeks.

Failides `areislah1.cpp` ja `areislah1.pas` toodud lahendused kasutavadki sellist algoritmi.

Eraldi tähelepanu nõuavad need ajavahemikud, mille sisse jääb kesköö. Et ka nendega rehkendamine õigesti välja tuleks, peab iga sellise vahemiku pikkusele ühe ööpäeva jagu minuteid juurde liitma. Ühe lennu jaoks võib nii koguneda kuni nelja ööpäeva jagu lisaminuteid:

- liini $A - B$ lennuki väljumise ja saabumise vahel võib olla südaöö;
- liini $A - B$ lennuki saabumise ja liini $B - C$ lennuki väljumise vahel võib olla südaöö;
- liini $A - B$ lennuki saabumise ja liini $B - C$ lennuki väljumise vahe võib olla liiga väike, siis on vaja lennujaamas üks ööpäev lisaks oodata;
- liini $B - C$ lennuki väljumise ja saabumise vahel võib olla südaöö.

Testid

1. $N_1 = 1$, $N_2 = 1$. Minimaalne lihtne test, mingeid erijuhte ei ole. 5 punkti.
2. $N_1 = 2$, $N_2 = 2$. Väike lihtne test, mingeid erijuhte ei ole. 5 punkti.
3. $N_1 = 3$, $N_2 = 3$. 30-minutise ümberistumisaja kontroll. 5 punkti.
4. $N_1 = 3$, $N_2 = 3$. 30-minutise ümberistumisaja kontroll, optimaalne lennuplaan on täpselt 30-minutise ajavaruga. 5 punkti.
5. $N_1 = 4$, $N_2 = 4$. Kõik lennud kestavad üle südaöö, suunal $A - B$ ainult üks lend alla 24 tunni. 5 punkti.
6. $N_1 = 2$, $N_2 = 1$. Ühel lennul võtab ümberistumine 10 minutit üle ööpäeva. 5 punkti.
7. $N_1 = 30$, $N_2 = 50$. Keskmine juhuslik test. 5 punkti.
8. $N_1 = 100$, $N_2 = 100$. Suur juhuslik test. 5 punkti.

Kokku 40 punkti.

6. Lennureis (edasijõudnute rühm)

Kõige lihtsam viis soodsaima reisi leidmiseks on vaadata läbi kõik lendudepaarid, kus üks lend on liinilt $A - B$ ja teine liinilt $B - C$. Selleks, et aegade võrdlemine selle läbivaatuse käigus lihtsam oleks, teisendame enne kõik ajad nädala algusest möödunud minutite arvudeks. Kuna nii lendude kui ka võimalike nädalapäevade arv on üsna väike, on kõige lihtsam mitmel nädalapäeval toimuvaid lende vaadelda mitme erineva lennuna, mis igaüks toimuvad ainult ühel nädalapäeval.

Failides `ereislah1.cpp` ja `ereislah1.pas` toodud lahendused kasutavadki sellist algoritmi.

Eraldi tähelepanu nõuavad need ajavahemikud, mille sisse jääb kesköö, ja eriti, kui see on pühapäeva ja esmaspäeva vaheline öö. Et ka nendega rehkendamine õigesti välja tuleks, peab iga sellise vahemiku pikkusele vastavalt vajadusele ööpäeva või nädala jagu minuteid juurde liitma:

- liini $A - B$ lennuki väljumise ja saabumise vahel võib olla südaöö, siis tuleb lennuajale ööpäev juure liita;
- liini $A - B$ lennuki saabumine võib olla nädala piires hiljem kui liini $B - C$ lennuki väljumine, siis tuleb lennujaama ooteajale nädal juurde liita;
- liini $A - B$ lennuki saabumise ja liini $B - C$ lennuki väljumise vahe võib olla liiga väike, siis tuleb lennujaama ooteajale nädal juurde liita;
- liini $B - C$ lennuki väljumise ja saabumise vahel võib olla südaöö, siis tuleb lennuajale ööpäev juure liita.

Testid

1. $N_1 = 2$, $N_2 = 2$. Väike lihtne test, mingeid erijuhte ei ole. 4 punkti.
2. $N_1 = 3$, $N_2 = 3$. 30-minutise ümberistumisaja kontroll. 4 punkti.
3. $N_1 = 3$, $N_2 = 3$. 30-minutise ümberistumisaja kontroll, optimaalne lennuplaan on täpselt 30-minutise ajavaruga. 4 punkti.
4. $N_1 = 4$, $N_2 = 4$. Kõik lennud kestavad üle südaöö, suunal $A - B$ ainult üks lend alla 24 tunni. 4 punkti.
5. $N_1 = 1$, $N_2 = 2$. Kontrollib nädalapäevadega arvutamist. 4 punkti.
6. $N_1 = 4$, $N_2 = 4$. Kontrollib üle nädalavahetuse lendamist. 4 punkti.
7. $N_1 = 2$, $N_2 = 1$. Ühel lennul võtab ümberistumine 10 minutit üle nädala. 4 punkti.
8. $N_1 = 15$, $N_2 = 10$. Väiksem juhuslik test, iga lend täpselt kaks korda nädalas. 4 punkti.
9. $N_1 = 15$, $N_2 = 10$. Maksimaalne juhuslik test, iga lend ainult ühel päeval. 4 punkti.
10. $N_1 = 100$, $N_2 = 100$. Maksimaalne juhuslik test. 4 punkti.

Kokku 40 punkti.