Lausnir á lokakeppnisdæmum

Bergur Snorrason, Atli FF

20. apríl 2022

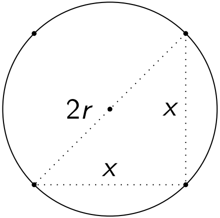
Skálagerð

- Þér er gefinn hringur með geisla r og þú átt að dreifa fjórum punktum jafnt á hringinn.

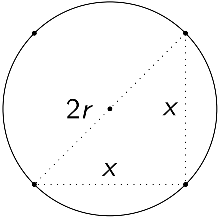
Skálagerð

- Þér er gefinn hringur með geisla r og þú átt að dreifa fjórum punktum jafnt á hringinn.
- Hver verður fjarlægðin milli aðliggjandi punkta?

Skálagerð

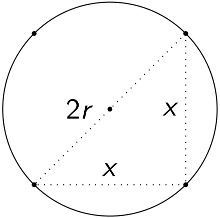


Skálagerð



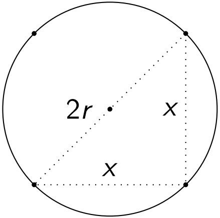
- Við vitum að þríhyrningurinn merktur með punktalínum er rétthyrndur (setning Þalesar).

Skálagerð



- Við vitum að þríhyrningurinn merktur með punktalínum er rétthyrndur (setning Þalesar).
- Svo $2x^2 = 4r^2$ (setning Pýþagorasar).

Skálagerð



- Við vitum að þríhyrningurinn merktur með punktalínum er rétthyrndur (setning Þalesar).
- Svo $2x^2 = 4r^2$ (setning Pýþagorasar).
- Svarið er því $x = r\sqrt{2}$.

DCC líkur

- Gefna teninga í DCC kerfinu og líkur p, hvað þarf að hækka annan teninginn mikið svo hann hafi p% vinningslíkur?

DCC líkur

- Líkurnar á að n hliða teningur sigri m hliða tening má hér einfaldlega reikna með tvöfaldri for-lykkju því tölurnar eru svo smáar.

DCC líkur

- Líkurnar á að n hliða teningur sigri m hliða tening má hér einfaldlega reikna með tvöfaldri for-lykkju því tölurnar eru svo smáar.
- Ytri lykkjan fer frá 1 til n, hin frá 1 til m og þegar ytri breytan er stærri (strangt!) hækkum við teljara um 1. Deilum með nm í lokin og fáum líkurnar.

DCC líkur

- Líkurnar á að n hliða teningur sigri m hliða tening má hér einfaldlega reikna með tvöfaldri for-lykkju því tölurnar eru svo smáar.
- Ytri lykkjan fer frá 1 til n, hin frá 1 til m og þegar ytri breytan er stærri (strangt!) hækkum við teljara um 1. Deilum með nm í lokin og fáum líkurnar.
- Þá er bara að gera þetta aftur og aftur þar til líkurnar eru stærri en eða jöfn (ekki strangt!) $p\%$, hækka tening um einn í keðjunni í einu.

DCC líkur

- Líkurnar á að n hliða teningur sigri m hliða tening má hér einfaldlega reikna með tvöfaldri for-lykkju því tölurnar eru svo smáar.
- Ytri lykkjan fer frá 1 til n, hin frá 1 til m og þegar ytri breytan er stærri (strangt!) hækkum við teljara um 1. Deilum með nm í lokin og fáum líkurnar.
- Þá er bara að gera þetta aftur og aftur þar til líkurnar eru stærri en eða jöfn (ekki strangt!) p%, hækka tening um einn í keðjunni í einu.
- Passa nákvæmni, betra jafnvel að nota almenn brot heldur en fleytitölur. Passa að ekki sé hægt að fara uppfyrir 30 hliðar.

Vatnskubbur

- Gefin heiltala $n \leq 10^{18}$, eru til heiltölur $a, b > 1$ þannig að $n = ab^2$?

Vatnskubbur

- Frumþáttum þannig að $n = p_1^{e_1} \dots p_m^{e_m}$.

Vatnskubbur

- Frumþáttum þannig að $n = p_1^{e_1} \dots p_m^{e_m}$.
- Tökum eftir að ef $e_1 = \cdots = e_m = 1$ þá er þetta ekki hægt.

Vatnskubbur

- Frumþáttum þannig að $n = p_1^{e_1} \dots p_m^{e_m}$.
- Tökum eftir að ef $e_1 = \cdots = e_m = 1$ þá er þetta ekki hægt.
- Ef $m = 1$ og $e_1 = 2$ þá er þetta heldur ekki hægt.

Vatnskubbur

- Frumþáttum þannig að $n = p_1^{e_1} \ldots p_m^{e_m}$.
- Tökum eftir að ef $e_1 = \cdots = e_m = 1$ þá er þetta ekki hægt.
- Ef $m = 1$ og $e_1 = 2$ þá er þetta heldur ekki hægt.
- Annars er þetta hægt.

Vatnskubbur

- Frumþáttum þannig að $n = p_1^{e_1} \dots p_m^{e_m}$.
- Tökum eftir að ef $e_1 = \cdots = e_m = 1$ þá er þetta ekki hægt.
- Ef $m = 1$ og $e_1 = 2$ þá er þetta heldur ekki hægt.
- Annars er þetta hægt.
- Þá er til j þannig að $e_j \geq 2$ svo við getum látið $b = p_j$.

Vatnskubbur

- Frumþáttum þannig að $n = p_1^{e_1} \dots p_m^{e_m}$.
- Tökum eftir að ef $e_1 = \cdots = e_m = 1$ þá er þetta ekki hægt.
- Ef $m = 1$ og $e_1 = 2$ þá er þetta heldur ekki hægt.
- Annars er þetta hægt.
- Þá er til j þannig að $e_j \geq 2$ svo við getum látið $b = p_j$.
- Við þurfum að passa að n er stór, svo við þurfum reiknirit Pollards til að lausnin verði nógu hröð.

Vatnskubbur

- Frumþáttum þannig að $n = p_1^{e_1} \ldots p_m^{e_m}$.
- Tökum eftir að ef $e_1 = \cdots = e_m = 1$ þá er þetta ekki hægt.
- Ef $m = 1$ og $e_1 = 2$ þá er þetta heldur ekki hægt.
- Annars er þetta hægt.
- Þá er til j þannig að $e_j \geq 2$ svo við getum látið $b = p_j$.
- Við þurfum að passa að n er stór, svo við þurfum reiknirit Pollards til að lausnin verði nógu hröð.
- Reiknirit Pollards er of hægt fyrir stóra frumtölu þar að auki, svo byrja þarf á að nota reiknirit Miller-Rabin.

Önnur lausn

- Einnig má vera aðeins sniðugur og sleppa öllu flottu reikniritunum.

Önnur lausn

- Einnig má vera aðeins sniðugur og sleppa öllu flottu reikniritunum.
- Það þarf aðeins að fjarlægja þættina úr n sem eru $\leq \sqrt[3]{n}$.

Önnur lausn

- Einnig má vera aðeins sniðugur og sleppa öllu flottu reikniritunum.
- Það þarf aðeins að fjarlægja þættina úr n sem eru $\leq \sqrt[3]{n}$.
- Eftirlátum restina af þessarri lausn sem æfingu fyrir lesanda.

Bíórugl

- Það eru $n \leq 10^{18}$ einstaklingar í bíó og þeir sitja allir í sömu röð og fylla akkúrat röðina.

Bíórugl

- Það eru $n \leq 10^{18}$ einstaklingar í bíó og þeir sitja allir í sömu röð og fylla akkúrat röðina.
- Í hlé fara allir á klóið og vilja svo sæti sem er í mesta lagi tveimur sætum frá upprunalega sætinu sínu.

Bíórugl

- Það eru $n \leq 10^{18}$ einstaklingar í bíó og þeir sitja allir í sömu röð og fylla akkúrat röðina.
- Í hlé fara allir á klóið og vilja svo sæti sem er í mesta lagi tveimur sætum frá upprunalega sætinu sínu.
- Á hversu marga vegu geta þeir sest aftur?

Bíórugl

- Við leysum þetta með því að finna rakningarvensl sem lýsa dæminu.

Bíórugl

- Við leysum þetta með því að finna rakningarvensl sem lýsa dæminu.
- Með því að skoða hvernig dæmið skiptist í smærri tilfelli (og handreikna grunntilfellin) fæst að

$$c_n = 14c_{n-1} + 2c_{n-3} - c_{n-5},$$

ef $n > 4$ og $c_0 = 1, c_1 = 1, c_2 = 2, c_3 = 6$ og $c_4 = 14$.

Bíórugl

- Við leysum þetta með því að finna rakningarvensl sem lýsa dæminu.
- Með því að skoða hvernig dæmið skiptist í smærri tilfelli (og handreikna grunntilfellin) fæst að

$$c_n = 14c_{n-1} + 2c_{n-3} - c_{n-5},$$

  ef $n > 4$ og $c_0 = 1, c_1 = 1, c_2 = 2, c_3 = 6$ og $c_4 = 14$.
- Við getum síðan notað fylkjamargföldun til að reikna c_n í logratíma.

Bíórugl

- Við leysum þetta með því að finna rakningarvensl sem lýsa dæminu.
- Með því að skoða hvernig dæmið skiptist í smærri tilfelli (og handreikna grunntilfellin) fæst að

$$c_n = 14c_{n-1} + 2c_{n-3} - c_{n-5},$$

  ef $n > 4$ og $c_0 = 1, c_1 = 1, c_2 = 2, c_3 = 6$ og $c_4 = 14$.
- Við getum síðan notað fylkjamargföldun til að reikna c_n í logratíma.
- Ef við viljum ekki reikna grunntilfellin í höndunum getum við notað tæmandi leit til þessa að finna þau.

Bíórugl

- Við leysum þetta með því að finna rakningarvensl sem lýsa dæminu.
- Með því að skoða hvernig dæmið skiptist í smærri tilfelli (og handreikna grunntilfellin) fæst að

  $$c_n = 14c_{n-1} + 2c_{n-3} - c_{n-5},$$

  ef $n > 4$ og $c_0 = 1, c_1 = 1, c_2 = 2, c_3 = 6$ og $c_4 = 14$.
- Við getum síðan notað fylkjamargföldun til að reikna c_n í logratíma.
- Ef við viljum ekki reikna grunntilfellin í höndunum getum við notað tæmandi leit til þessa að finna þau.
- Við getum líka fundið stuðlana með Gauss-Jordan eyðingu.

Réttur krappi er rangur

- Gefnir eru $n \leq 3\,000$ punktar í plani.

Réttur krappi er rangur

- Gefnir eru $n \leq 3\,000$ punktar í plani.
- Hversu margar þrenndir í punktasafninu mynda rétthyrndan þríhyrning?

Réttur krappi er rangur

- Það er lítið mál að skoða allar þrenndir punkta, en sú lausn er $\mathcal{O}(n^3)$ sem er of hægt.

Réttur krappi er rangur

- Það er lítið mál að skoða allar þrenndir punkta, en sú lausn er $\mathcal{O}(n^3)$ sem er of hægt.
- Veljum einhvern punkt sem vendipunkt og skoðum allar línur sem liggja gegnum vendipunktinn og einhvern annan punkt í safninu.

Réttur krappi er rangur

- Það er lítið mál að skoða allar þrenndir punkta, en sú lausn er $\mathcal{O}(n^3)$ sem er of hægt.
- Veljum einhvern punkt sem vendipunkt og skoðum allar línur sem liggja gegnum vendipunktinn og einhvern annan punkt í safninu.
- Ef tvær línur skerast í réttu horni þá svara þær til þrenndar í punktasafninu sem myndar rétthyrndan þríhyrning.

Réttur krappi er rangur

- Það er lítið mál að skoða allar þrenndir punkta, en sú lausn er $\mathcal{O}(n^3)$ sem er of hægt.
- Veljum einhvern punkt sem vendipunkt og skoðum allar línur sem liggja gegnum vendipunktinn og einhvern annan punkt í safninu.
- Ef tvær línur skerast í réttu horni þá svara þær til þrenndar í punktasafninu sem myndar rétthyrndan þríhyrning.
- Við getum fundið, fyrir tiltekna línu, hversu margar línur hún sker undir réttu horni með helmingunarleit (tveimur leitum reyndar) eða gagngrindum á borða við leitartré (`multiset<...>`) eða hakkatöflu (`unordered_map<...>`).

Réttur krappi er rangur

- Það er lítið mál að skoða allar þrenndir punkta, en sú lausn er $\mathcal{O}(n^3)$ sem er of hægt.
- Veljum einhvern punkt sem vendipunkt og skoðum allar línur sem liggja gegnum vendipunktinn og einhvern annan punkt í safninu.
- Ef tvær línur skerast í réttu horni þá svara þær til þrenndar í punktasafninu sem myndar rétthyrndan þríhyrning.
- Við getum fundið, fyrir tiltekna línu, hversu margar línur hún sker undir réttu horni með helmingunarleit (tveimur leitum reyndar) eða gagngrindum á borða við leitartré (`multiset<...>`) eða hakkatöflu (`unordered_map<...>`).
- Endurtökum svo þannig að allir punktar verði vendipunktar og styttum út endurtekningar.

Réttur krappi er rangur

- Það er lítið mál að skoða allar þrenndir punkta, en sú lausn er $\mathcal{O}(n^3)$ sem er of hægt.
- Veljum einhvern punkt sem vendipunkt og skoðum allar línur sem liggja gegnum vendipunktinn og einhvern annan punkt í safninu.
- Ef tvær línur skerast í réttu horni þá svara þær til þrenndar í punktasafninu sem myndar rétthyrndan þríhyrning.
- Við getum fundið, fyrir tiltekna línu, hversu margar línur hún sker undir réttu horni með helmingunarleit (tveimur leitum reyndar) eða gagngrindum á borða við leitartré (`multiset<...>`) eða hakkatöflu (`unordered_map<...>`).
- Endurtökum svo þannig að allir punktar verði vendipunktar og styttum út endurtekningar.
- Þessi lausn er $\mathcal{O}(n^2 \log n)$.

Önnur lausn

- Einnig má nýta sér að þetta séu allt heiltölur. Ef við erum með línu gegnum $(0, 0)$ og tvo punkta verða hnit annars punktsins að vera margfeldi af hnitum hins.

Önnur lausn

- Einnig má nýta sér að þetta séu allt heiltölur. Ef við erum með línu gegnum $(0, 0)$ og tvo punkta verða hnit annars punktsins að vera margfeldi af hnitum hins.
- Veljum þá einn vendipunkt í einu og styttum út stærsta samdeili hnita allra punkta til að fá punktasafn, höldum utan um hvað það eru mörg af hverjum punkti því við fáum mögulega endurtekningar.

Önnur lausn

- Einnig má nýta sér að þetta séu allt heiltölur. Ef við erum með línu gegnum $(0, 0)$ og tvo punkta verða hnit annars punktsins að vera margfeldi af hnitum hins.
- Veljum þá einn vendipunkt í einu og styttum út stærsta samdeili hnita allra punkta til að fá punktasafn, höldum utan um hvað það eru mörg af hverjum punkti því við fáum mögulega endurtekningar.
- Svo fyrir hvern punkt skoðum við bara hvað það eru margir af honum og af honum snúið um $\pi/2$, leggjum það við niðurstöðu.

Önnur lausn

- Einnig má nýta sér að þetta séu allt heiltölur. Ef við erum með línu gegnum $(0, 0)$ og tvo punkta verða hnit annars punktsins að vera margfeldi af hnitum hins.
- Veljum þá einn vendipunkt í einu og styttum út stærsta samdeili hnita allra punkta til að fá punktasafn, höldum utan um hvað það eru mörg af hverjum punkti því við fáum mögulega endurtekningar.
- Svo fyrir hvern punkt skoðum við bara hvað það eru margir af honum og af honum snúið um $\pi/2$, leggjum það við niðurstöðu.
- Þessi lausn er $\mathcal{O}(n^2 \log(w))$ þar sem w er stærsta leyfilega hnit talnanna, sem gengur einnig.

Leiðinda rigning

- Finna á leiðina heim fyrir Atla sem bleytir hann sem minnst. Höfum net með $n \leq 5 \cdot 10^4$ hnúta, $m \leq 10^5$ leggi og svo $q \leq 5 \cdot 10^4$ fyrirspurnar. Þær biðja annað hvort um að breyta hvort hnútur sé strætóstöð eða að finna hvaða stöð er næst gefnum hnút.

Drög að lausn

- Fjarlægðin er ekki summa vigtanna, heldur bara hæsta vigtin sem kemur fyrir á leiðinni. Við getum því hent öllum leggjum sem eru ekki í minnsta spannandi tré netsins.

Drög að lausn

- Fjarlægðin er ekki summa vigtanna, heldur bara hæsta vigtin sem kemur fyrir á leiðinni. Við getum því hent öllum leggjum sem eru ekki í minnsta spannandi tré netsins.
- Þá erum við með tré. Ímyndum okkur að við viljum reikna fjarlægð í næstu stöð fyrir alla hnúta í byrjun. Getum gert þetta með reikniriti Dijkstra, setjum allar strætóstöðvar sem fjarlægð 0 í byrjun.

Drög að lausn

- Fjarlægðin er ekki summa vigtanna, heldur bara hæsta vigtin sem kemur fyrir á leiðinni. Við getum því hent öllum leggjum sem eru ekki í minnsta spannandi tré netsins.
- Þá erum við með tré. Ímyndum okkur að við viljum reikna fjarlægð í næstu stöð fyrir alla hnúta í byrjun. Getum gert þetta með reikniriti Dijkstra, setjum allar strætóstöðvar sem fjarlægð 0 í byrjun.
- En við getum ekki uppfært þetta nógu hratt. Hvað ef við viljum skoða fjarlægð í eina tiltekna stöð hratt?

Drög að lausn

- Fjarlægðin er ekki summa vigtanna, heldur bara hæsta vigtin sem kemur fyrir á leiðinni. Við getum því hent öllum leggjum sem eru ekki í minnsta spannandi tré netsins.
- Þá erum við með tré. Ímyndum okkur að við viljum reikna fjarlægð í næstu stöð fyrir alla hnúta í byrjun. Getum gert þetta með reikniriti Dijkstra, setjum allar strætóstöðvar sem fjarlægð 0 í byrjun.
- En við getum ekki uppfært þetta nógu hratt. Hvað ef við viljum skoða fjarlægð í eina tiltekna stöð hratt?
- Setjum upp LCA töflu fyrir tréð! Þá getum við fundið hámarksvigtina milli upphafspunkts og að strætóstöð í logratíma.

Drög að lausn

- Fjarlægðin er ekki summa vigtanna, heldur bara hæsta vigtin sem kemur fyrir á leiðinni. Við getum því hent öllum leggjum sem eru ekki í minnsta spannandi tré netsins.
- Þá erum við með tré. Ímyndum okkur að við viljum reikna fjarlægð í næstu stöð fyrir alla hnúta í byrjun. Getum gert þetta með reikniriti Dijkstra, setjum allar strætóstöðvar sem fjarlægð 0 í byrjun.
- En við getum ekki uppfært þetta nógu hratt. Hvað ef við viljum skoða fjarlægð í eina tiltekna stöð hratt?
- Setjum upp LCA töflu fyrir tréð! Þá getum við fundið hámarksvigtina milli upphafspunkts og að strætóstöð í logratíma.
- Hvernig má nú sameina þetta tvennt til að fá skikkanlega tímaflækju?

Rótarþáttun

- Við skiptum fyrirspurnunum í $\sqrt{q}$ fötur. Í byrjuninni á hverri fötu reiknum við allar fjarlægðir í stöðvar sem munu ekki breytast í þessarri fötu með því að nota Dijkstra.

Rótarþáttun

- Við skiptum fyrirspurnunum í $\sqrt{q}$ fötur. Í byrjuninni á hverri fötu reiknum við allar fjarlægðir í stöðvar sem munu ekki breytast í þessarri fötu með því að nota Dijkstra.
- Löbbum svo í gegnum fötuna. Flettum upp gildi í Dijkstra niðurstöðum fyrir hverja fyrirspurn, en höldum einnig utan um lista fyrir allar breyttar stöðvar. Við reiknum fjarlægðirnar í þær allar til viðbótar með LCA og tökum besta gildið. Þessi listi verður aldrei lengri en $\sqrt{q}$ því hann getur aðeins breyst um eitt stak í hverri fyrirspurn.

Rótarþáttun

- Við skiptum fyrirspurnunum í $\sqrt{q}$ fötur. Í byrjuninni á hverri fötu reiknum við allar fjarlægðir í stöðvar sem munu ekki breytast í þessarri fötu með því að nota Dijkstra.
- Löbbum svo í gegnum fötuna. Flettum upp gildi í Dijkstra niðurstöðum fyrir hverja fyrirspurn, en höldum einnig utan um lista fyrir allar breyttar stöðvar. Við reiknum fjarlægðirnar í þær allar til viðbótar með LCA og tökum besta gildið. Þessi listi verður aldrei lengri en $\sqrt{q}$ því hann getur aðeins breyst um eitt stak í hverri fyrirspurn.
- Reiknum Dijkstra $\sqrt{q}$ sinnum, það tekur samtals $\mathcal{O}(\sqrt{q}n\log(n))$ tíma. Reiknum LCA töflu í byrjun í $\mathcal{O}(n\log(n))$ tíma. Flettum upp í henni fyrir hvert stak listans og í hverri fyrirspurn, það tekur $\mathcal{O}(q\sqrt{q}\log(n))$. Ef við reiknum upp úr þessu sést að þetta allt saman er undir tímamörkum.