Lausnir á völdum dæmum

Dæmahöfundar

- Arnar Bjarni Arnarson
- Atli Fannar Franklín
- Bernhard Linn Hilmarsson
- Bjarki Ágúst Guðmundsson
- Bjarni Dagur Thor Kárason
- Elvar Árni Bjarnason
- Hannes Kristján Hannesson
- Samúel Arnar Hafsteinsson
- Unnar Freyr Erlendsson

Dæmayfirferð

- Arnar Páll Jóhannsson
- Álfur Birkir Bjarnason
- Ásþór Björnsson
- Bergur Snorrason
- Guðni Nathan Gunnarsson
- Sesar Hersisson

Ball

Dæmið

Pör eru skráð á ball þar sem hver manneskja var skráð einu sinni. Einu pari var síðan bætt við á skráningarlistann. Skrifaðu út parið sem var bætt við.

Sýnidæmi

Inntak:

```
10
1 2
3 5
4 8
6 7
4 7
9 10
```

Úttak:

```
4 7
```

Ball

Einföld lausn - $N \leq 200$

Viljum finna parið sem á að fjarlægja þannig engin tala komi tvisvar fyrir

Ball

Einföld lausn - $N \leq 200$

Viljum finna parið sem á að fjarlægja þannig engin tala komi tvisvar fyrir

Prófum alla möguleika, fyrir hvert gildi á i þannig $1 \leq i \leq \frac{n}{2} + 1$.

Ball

Einföld lausn - $N \leq 200$

Viljum finna parið sem á að fjarlægja þannig engin tala komi tvisvar fyrir

Prófum alla möguleika, fyrir hvert gildi á i þannig $1 \leq i \leq \frac{n}{2} + 1$.

- Látum eins og stak á vísi i sé ekki tekið með.

Ball

Einföld lausn - $N \leq 200$

Viljum finna parið sem á að fjarlægja þannig engin tala komi tvisvar fyrir

Prófum alla möguleika, fyrir hvert gildi á i þannig $1 \leq i \leq \frac{n}{2} + 1$.

- Látum eins og stak á vísi i sé ekki tekið með.
- Gáum hvort eitthvað gildi komi fyrir í tveimur mismunandi pörum.

Ball

Einföld lausn - $N \leq 200$

Viljum finna parið sem á að fjarlægja þannig engin tala komi tvisvar fyrir

Prófum alla möguleika, fyrir hvert gildi á i þannig $1 \leq i \leq \frac{n}{2} + 1$.

- Látum eins og stak á vísi i sé ekki tekið með.
- Gáum hvort eitthvað gildi komi fyrir í tveimur mismunandi pörum.
- Ef ekkert gildi kemur fyrir í tveim mismunandi pörum þá fundum við svarið.

Ball

Kóði

```
n = int(input())
pairs = [tuple(map(int, input().split())) for i in range(n//2+1)]
for i in range(len(pairs)):
    conflict = False
    for j in range(len(pairs)):
        if i == j:
            continue
        for k in range(j+1,len(pairs)):
            if i == k:
                continue
            if (pairs[j][0] in pairs[k] or
                pairs[j][1] in pairs[k]):
                conflict = True
    if conflict == False:
        print("{} {}".format(*sorted(pairs[i])))
        break
```

Tímaflækjan er $\mathcal{O}(n^3)$.

Ball

Bætt lausn - $N \leq 5000$

Óþarfi að bera saman sérhvert par við sérhvert annað par

Ball

Bætt lausn - $N \leq 5000$

Óþarfi að bera saman sérhvert par við sérhvert annað par

Höldum frekar utan um stök sem við höfum séð áður með bool fylki

Ball

Bætt lausn - $N \leq 5000$

- Óþarfi að bera saman sérhvert par við sérhvert annað par
- Höldum frekar utan um stök sem við höfum séð áður með bool fylki
- Frumstillum bool fylkið þannig öll stök séu óséð, með False

Ball

Bætt lausn - $N \leq 5000$

- Óþarfi að bera saman sérhvert par við sérhvert annað par
- Höldum frekar utan um stök sem við höfum séð áður með bool fylki
- Frumstillum bool fylkið þannig öll stök séu óséð, með False
- Rennum í gegnum öll pör og gáum hvort við höfum séð annað stak parsins

Ball

Bætt lausn - $N \leq 5000$

- Óþarfi að bera saman sérhvert par við sérhvert annað par
- Höldum frekar utan um stök sem við höfum séð áður með bool fylki
- Frumstillum bool fylkið þannig öll stök séu óséð, með False
- Rennum í gegnum öll pör og gáum hvort við höfum séð annað stak parsins
- Ef stak hefur sést áður þá er ekki svar hér, annars merkjum við stökin í bool fylkinu með True

Ball

Kóði

```
n = int(input())
pairs = [tuple(map(int, input().split())) for i in range(n//2+1)]
for i in range(len(pairs)):
    s = [False for _ in range(n+1)]
    for j in range(len(pairs)):
        if i == j:
            continue
        a,b = pairs[j]
        if s[a] or s[b]:
            break
        s[a], s[b] = True, True
    else:
        print("{} {}".format(*sorted(pairs[i])))
        break
```

Tímaflækjan er $\mathcal{O}(n^2)$.

Ball

Full lausn

Óþarfi að prófa að fjarlægja sérhvert par

Ball

Full lausn

Óþarfi að prófa að fjarlægja sérhvert par

Í upprunalega listanum kom hvert stak fyrir einu sinni

Ball

Full lausn

Óþarfi að prófa að fjarlægja sérhvert par

Í upprunalega listanum kom hvert stak fyrir einu sinni

Stökin í parinu sem var bætt við koma því fyrir tvisvar

Ball

Kóði

```python
from collections import Counter
n = int(input())
pairs = [tuple(map(int, input().split())) for i in range(n//2+1)]
c = Counter()
for a,b in pairs:
    c[a] += 1
    c[b] += 1
a, b = c.most_common(2)
a,b = a[0], b[0]
if b < a:
    a, b = b, a
print("{} {}".format(a, b))
```

Tímaflækjan er $\mathcal{O}(n)$.

Samlokur

Dæmið

Þú átt n samlokur sem hafa allar gæðagildi ásamt því að renna út á tilteknum tíma. Þú borðar tvær samlokur á dag og samtals gæðigildi þeirra þarf að vera meira en 8 og þú vilt ekki borða samlokur með gæðagildi lægra en 4. Geturðu fundið út í hvaða röð skal borða samlokurnar til að láta þær endast í k daga.

Samlokur

Lausn

- Hugsum þetta aftur á bak. Í staðinn fyrir að samlokur renni út á tilteknum degi þá verða þær aðgengilegar á þeim degi.
- Skiptum samlokunum í þrjá flokka: þær sem hafa gæðagildi minna en 4, þær sem hafa gæðagildi 4 og þær sem hafa gæðagildi meira en 4.
- Við viljum aldrei borða samloku í fyrsta flokkinum. Við viljum alltaf gera eitt af tvennu:
  - Borða tvær samlokur úr þriðja flokknum.
  - Borða eina samloku í öðrum flokknum og eina samloku í þriðja flokknum.

Samlokur

Lausn

- Ein leið til að leysa þetta er með kvikri bestun.
- Látum fallið $f(j, x, y)$ tákna hvort við getum enst fyrstu j dagana ef við höfum, á j-ta degi, x samlokur í öðrum flokknum og y samlokur í þriðja flokknum.
- Við fáum þá að $f(j, x, y)$ er satt þá og því aðeins að $f(j-1, x-1, y-1)$ eða $f(j-1, x, y-2)$ sé satt.
- Þessi lausn er $O(k \cdot n^2)$ og fær 40 stig. En nú ákvarðast y í $f(j, x, y)$ útfrá j og x, svo það er hægt gera lausnina $O(n \cdot k)$ og fá fyrir það 60 stig.
- En hvernig fáum við 100 stig?
- Við getum leyst þetta gráðugt. Við borðum alltaf samloku með gæðagildi 4 ef við getum.
- Þá fáum við lausn sem er $O(n + k)$ sem gefur 100 stig.

Dæmið

Gefið er net á n hnútum með m leggjum sem hafa hvert þrívíðan vigur sem vigt. Eru til tvær leiðir milli staða sem hafa ólíka summu vigta?

Ríkjafræði

Dæmið

Gefið er net á n hnútum með m leggjum sem hafa hvert þrívíðan vigur sem vigt. Eru til tvær leiðir milli staða sem hafa ólíka summu vigta?

Lausn

- Ef við festum einhvern einn hnút sem grunnpunkt og segjum að hann sé $(0, 0, 0)$ getum við reiknað út hvar aðrir hnútar eru útfrá vigtum leggjanna.
- Ef allar leiðir gefa sömu summuna ætti þetta að gefa gildi á öllum hnútum sem samræmist öllum leggjum.
- Þurfum samt að passa að velja slíkan grunnpunkt í hverjum samhengisþætti ef netið er ekki samanhangandi.

Ríkjafræði

Lausn

- Það dugar því að keyra eitt DFS í hverjum samhengisþætti. Notum leggina í DFS spanntrénu til að setja gildi hnútanna. Skilum svo false ef einhver leggur utan spanntrésins samræmist ekki útreiknuðum gildum.
- Þetta er $\mathcal{O}(n + m)$ sem er nógu hratt og gefur því 100 stig.

Kolkrabbaleikarnir

Dæmið

m manns eru að reyna komast yfir n glerbrúarskref. Í hverju skrefi eru helmingslíkur að fremsti maður deyji. Hver er væntur fjöldi manns sem kemst alla leið yfir?

Kolkrabbaleikarnir

Dæmið

m manns eru að reyna komast yfir n glerbrúarskref. Í hverju skrefi eru helmingslíkur að fremsti maður deyji. Hver er væntur fjöldi manns sem kemst alla leið yfir?

Lausn

- Látum $f(n, m)$ vera svarið fyrir n, m. Sjáum að ef engin skref eru eftir lifa allir af og ef engir eru eftir lifir enginn af. Þ.e.a.s. $f(0, m) = m$ og $f(n, 0) = 0$.
- Fyrir önnur gildi á n, m þá mun skrefum áfallt fækka um einn, í helming skipta fækkar mönnum um einn þar að auki. Þ.e. $f(n, m) = f(n - 1, m)/2 + f(n - 1, m - 1)/2$.

Kolkrabbaleikarnir

Lausn

- Endurkvæm lausn sem reiknar þetta fær 60 stig. Til að fá fleiri þarf að gera þetta hraðar.
- Ef við byrjum í (n, m) munum við fara endurkvæmt í $(n-2, m-2)$ sex sinnum og reikna sama gildið í hvert sinn. Ef við geymum öll útreiknuðu $f(n, m)$ gildin í töflu og reiknum þau bara einu sinni hvert verður lausnin mun hraðari, þ.e. $\mathcal{O}(nm)$. Þetta gefur 80 stig.

Kolkrabbaleikarnir

Lausn

- Loks skoðum við hvað þarf til að fá 100 stig. Sjáum að endurkvæmu köllin enda alltaf í $f(0, k)$ og vinna sig svo aftur upp. Enn fremur tekur alltaf n skref til að komast þangað. Þar með bætist $k/2^n$ við svarið í hvert sinn sem við endum í $f(0, k)$. En hversu oft endum við í $f(0, k)$ ef við byrjum í (n, m)?
- Tökum n skref og viljum að $m - k$ þeirra séu tilfelli þar sem einhver deyr. Það eru því $\binom{n}{m-k}$ leiðir til þess.
- Svarið er því

$$\frac{1}{2^n} \sum_{k=0}^{m} k \binom{n}{m-k}$$

sem reikna má nógu hratt til að fá 100 stig.

Pizzastrengur

Dæmið

Gefin er lengd leyniorðs sem inniheldur einungis stafina {“`P`”, “`I`”, “`Z`”, “`A`”}. Við getum spurt hvort orð sé forskeyti af leyniorðinu og viljum finna leyniorðið í sem fæstum giskum.

Sýnidæmi

Pizzastrengur

Dæmið

Gefin er lengd leyniorðs sem inniheldur einungis stafina {“`P`”, “`I`”, “`Z`”, “`A`”}. Við getum spurt hvort orð sé forskeyti af leyniorðinu og viljum finna leyniorðið í sem fæstum giskum.

Sýnidæmi

Inntak	Úttak

Pizzastrengur

Dæmið

Gefin er lengd leyniorðs sem inniheldur einungis stafina {“`P`”, “`I`”, “`Z`”, “`A`”}. Við getum spurt hvort orð sé forskeyti af leyniorðinu og viljum finna leyniorðið í sem fæstum giskum.

Sýnidæmi

Inntak	Úttak
5	

Pizzastrengur

Dæmið

Gefin er lengd leyniorðs sem inniheldur einungis stafina {“P”, “I”, “Z”, “A”}. Við getum spurt hvort orð sé forskeyti af leyniorðinu og viljum finna leyniorðið í sem fæstum giskum.

Sýnidæmi

Inntak	Úttak
5	
	Z

Pizzastrengur

Dæmið

Gefin er lengd leyniorðs sem inniheldur einungis stafina {“`P`”, “`I`”, “`Z`”, “`A`”}. Við getum spurt hvort orð sé forskeyti af leyniorðinu og viljum finna leyniorðið í sem fæstum giskum.

Sýnidæmi

Inntak	Úttak
5	
	Z
0	

Pizzastrengur

Dæmið

Gefin er lengd leyniorðs sem inniheldur einungis stafina {“`P`”, “`I`”, “`Z`”, “`A`”}. Við getum spurt hvort orð sé forskeyti af leyniorðinu og viljum finna leyniorðið í sem fæstum giskum.

Sýnidæmi

Inntak	Úttak
5	
	Z
0	
	PIZ

Pizzastrengur

Dæmið

Gefin er lengd leyniorðs sem inniheldur einungis stafina {“`P`”, “`I`”, “`Z`”, “`A`”}. Við getum spurt hvort orð sé forskeyti af leyniorðinu og viljum finna leyniorðið í sem fæstum giskum.

Sýnidæmi

Inntak	Úttak
5	
	Z
0	
	PIZ
1	

Pizzastrengur

Dæmið

Gefin er lengd leyniorðs sem inniheldur einungis stafina {“`P`”, “`I`”, “`Z`”, “`A`”}. Við getum spurt hvort orð sé forskeyti af leyniorðinu og viljum finna leyniorðið í sem fæstum giskum.

Sýnidæmi

Inntak	Úttak
5	
	Z
0	
	PIZ
1	
	PIZZA

Pizzastrengur

Dæmið

Gefin er lengd leyniorðs sem inniheldur einungis stafina {“P”, “I”, “Z”, “A”}. Við getum spurt hvort orð sé forskeyti af leyniorðinu og viljum finna leyniorðið í sem fæstum giskum.

Sýnidæmi

Inntak	Úttak
5	
	Z
0	
	PIZ
1	
	PIZZA
2	

Pizzastrengur

Einföld lausn - $\leq 4N$ gisk

Ef við erum með rétt forskeyti getum við fundið næsta staf með því að prófa að bæta hverjum staf við forskeytið.

Pizzastrengur

Einföld lausn - $\leq 4N$ gisk

Ef við erum með rétt forskeyti getum við fundið næsta staf með því að prófa að bæta hverjum staf við forskeytið.

Fyrst prófum við að bæta “`P`” við.

Pizzastrengur

Einföld lausn - $\leq 4N$ gisk

Ef við erum með rétt forskeyti getum við fundið næsta staf með því að prófa að bæta hverjum staf við forskeytið.

Fyrst prófum við að bæta "`P`" við.

Næst prófum við að bæta "`I`" við.

Pizzastrengur

Einföld lausn - $\leq 4N$ gisk

Ef við erum með rétt forskeyti getum við fundið næsta staf með því að prófa að bæta hverjum staf við forskeytið.

Fyrst prófum við að bæta “`P`” við.

Næst prófum við að bæta “`I`” við.

Svo prófum við að bæta “`Z`” við.

Pizzastrengur

Einföld lausn - $\leq 4N$ gisk

Ef við erum með rétt forskeyti getum við fundið næsta staf með því að prófa að bæta hverjum staf við forskeytið.

Fyrst prófum við að bæta “`P`” við.

Næst prófum við að bæta “`I`” við.

Svo prófum við að bæta “`Z`” við.

Loks prófum við að bæta “`A`” við.

Pizzastrengur

Einföld lausn - $\leq 4N$ gisk

Ef við erum með rétt forskeyti getum við fundið næsta staf með því að prófa að bæta hverjum staf við forskeytið.

Fyrst prófum við að bæta “`P`” við.

Næst prófum við að bæta “`I`” við.

Svo prófum við að bæta “`Z`” við.

Loks prófum við að bæta “`A`” við.

Þegar réttur stafur er fundinn er honum bætt við forskeytið.

Pizzastrengur

Einföld lausn - $\leq 4N$ gisk

Ef við erum með rétt forskeyti getum við fundið næsta staf með því að prófa að bæta hverjum staf við forskeytið.

Fyrst prófum við að bæta “`P`” við.

Næst prófum við að bæta “`I`” við.

Svo prófum við að bæta “`Z`” við.

Loks prófum við að bæta “`A`” við.

Þegar réttur stafur er fundinn er honum bætt við forskeytið.

Þetta tekur í versta falli $4N$ gisk.

Pizzastrengur

Einföld lausn - $\leq 4N$ gisk

Ef við erum með rétt forskeyti getum við fundið næsta staf með því að prófa að bæta hverjum staf við forskeytið.

Fyrst prófum við að bæta “`P`” við.

Næst prófum við að bæta “`I`” við.

Svo prófum við að bæta “`Z`” við.

Loks prófum við að bæta “`A`” við.

Þegar réttur stafur er fundinn er honum bætt við forskeytið.

Þetta tekur í versta falli $4N$ gisk.

Hvernig getum við gert betur?

Pizzastrengur

Bætt lausn - $\leq 3N + 1$ gisk

Við erum með rétt forskeyti og erum að finna næsta staf.

Pizzastrengur

Bætt lausn - $\leq 3N + 1$ gisk

Við erum með rétt forskeyti og erum að finna næsta staf.

Prófum að bæta “`P`” við en það er ekki rétt.

Pizzastrengur

Bætt lausn - $\leq 3N + 1$ gisk

Við erum með rétt forskeyti og erum að finna næsta staf.

Prófum að bæta “`P`” við en það er ekki rétt.

Prófum að bæta “`I`” við en það er ekki rétt.

Pizzastrengur

Bætt lausn - $\leq 3N + 1$ gisk

Við erum með rétt forskeyti og erum að finna næsta staf.

Prófum að bæta "`P`" við en það er ekki rétt.

Prófum að bæta "`I`" við en það er ekki rétt.

Prófum að bæta "`Z`" við en það er ekki rétt.

Pizzastrengur

Bætt lausn - $\leq 3N+1$ gisk

Við erum með rétt forskeyti og erum að finna næsta staf.

Prófum að bæta "`P`" við en það er ekki rétt.

Prófum að bæta "`I`" við en það er ekki rétt.

Prófum að bæta "`Z`" við en það er ekki rétt.

Þurfum ekki að prófa bæta "`A`" við. "`A`" verður að vera næsti stafur.

Pizzastrengur

Bætt lausn - $\leq 3N + 1$ gisk

Við erum með rétt forskeyti og erum að finna næsta staf.

Prófum að bæta "`P`" við en það er ekki rétt.

Prófum að bæta "`I`" við en það er ekki rétt.

Prófum að bæta "`Z`" við en það er ekki rétt.

Þurfum ekki að prófa bæta "`A`" við. "`A`" verður að vera næsti stafur.

Þetta tekur í versta falli $3N + 1$ gisk.

Pizzastrengur

Bætt lausn - $\leq 2.75N$ gisk

Gallinn við að giska alltaf á næsta staf í sömu röð er að það er auðvelt að búa til leyniorð sem þarf $4N$ gisk.

Pizzastrengur

Bætt lausn - $\leq 2.75N$ gisk

Gallinn við að giska alltaf á næsta staf í sömu röð er að það er auðvelt að búa til leyniorð sem þarf $4N$ gisk.

Ef við giskum á næsta staf í handahófskenndri röð þá er væntifjöldi giska

$$\frac{1+2+3+4}{4}N = 2.5N$$

Pizzastrengur

Full lausn - $\leq 2.45N$ gisk

Ef við giskum á næsta staf í handahófskenndri röð

Pizzastrengur

Full lausn - $\leq 2.45N$ gisk

Ef við giskum á næsta staf í handahófskenndri röð og sleppum því að giska á fjórða stafinn

Pizzastrengur

Full lausn - $\leq 2.45N$ gisk

Ef við giskum á næsta staf í handahófskenndri röð
og sleppum því að giska á fjórða stafinn
þá er væntifjöldi giska

$$\frac{1+2+3+3}{4}N = 2.25N$$

Rúnaheimur

Dæmið

Erum með svokallað rennipúsl sem við viljum leysa. Hæsta talan táknar tóma plássið og svo má renna hinum reitunum með aðgerðunum U, D, L og R.

Sýnidæmi

Inntak:

```
2 2
2 3
1 4
```

Úttak:

```
RDLURDLU
```

Rúnaheimur

Lausnir

- Margar mismunandi og misgóðar lausnaraðferðir.
- Fyrir 10×10 leikborð er tala með 156 tölustafi.
- 46663107721972076340849619428133350245357984132190810734296481947608799996614957804470731988078259143126
848960413611879125592605458432000000000000000000000000

Rúnaheimur

Lausnir

- Margar mismunandi og misgóðar lausnaraðferðir.
- Hóp 1 má leysa með því að framkvæma “RDLU” núll til þrisvar sinnum.
- Fyrir 10×10 leikborð er tala með 156 tölustafi.
- 4666310772197207634084961942813335024535798413219081
  0734296481947608799996614957804470731988078259143126
  8489604136118791255926054584320000000000000000000000

Rúnaheimur

Lausnir

- Margar mismunandi og misgóðar lausnaraðferðir.
- Hóp 1 má leysa með því að framkvæma “RDLU” núll til þrisvar sinnum.
- Hægt er að nota breiddarleit (BFS) eða dýptarleit (DFS) til að finna skrefin að leystu leikborði sem ætti að ná nokkrum hópum.
- Fyrir 10×10 leikborð er tala með 156 tölustafi.
- 466631077219720763408496194281333502453579841321908107342964819476087999966149578044707319880782591431268489604136118791255926054584320000000000000000000000

Rúnaheimur

Lausnir

- Margar mismunandi og misgóðar lausnaraðferðir.
- Hóp 1 má leysa með því að framkvæma “RDLU” núll til þrisvar sinnum.
- Hægt er að nota breiddarleit (BFS) eða dýptarleit (DFS) til að finna skrefin að leystu leikborði sem ætti að ná nokkrum hópum.
- Fjöldi mismunandi leikborða af stærð $n \times m$ er hinsvegar $\frac{(nm-1)!}{2}$ þannig það er of hægt fyrir stærri hópa.
- Fyrir 10×10 leikborð er tala með 156 tölustafi.
- 466631077219720763408496194281333502453579841321908107342964819476087999966149578044707319880782591431268489604136118791255926054584320000000000000000000000

Rúnaheimur

Lausnir

- Margar mismunandi og misgóðar lausnaraðferðir.
- Hóp 1 má leysa með því að framkvæma “RDLU” núll til þrisvar sinnum.
- Hægt er að nota breiddarleit (BFS) eða dýptarleit (DFS) til að finna skrefin að leystu leikborði sem ætti að ná nokkrum hópum.
- Fjöldi mismunandi leikborða af stærð $n \times m$ er hinsvegar $\frac{(nm-1)!}{2}$ þannig það er of hægt fyrir stærri hópa.
- Til dæmis er fjöldi mismunandi staða fyrir 4×4 er $653\,837\,184\,000$ sem tæki a.m.k. klukkutíma að fara í gegnum með þessum leitaraðferðum.
- Fyrir 10×10 leikborð er tala með 156 tölustafi.
- 466631077219720763408496194281333502453579841321908107342964819476087999966149578044707319880782591431268489604136118791255926054584320000000000000000000000000

Rúnaheimur

Lausnir frh.

- Viljum forðast að skoða of margar stöður, sérstaklega þær sem færa okkur fjær réttu lausninni.

Lausnir frh.

- Viljum forðast að skoða of margar stöður, sérstaklega þær sem færa okkur fjær réttu lausninni.
- Til eru tvær aðferðir sem eru svipaðar fyrri leitaraðferðum: A* og IDA*.

Rúnaheimur

Lausnir frh.

- Viljum forðast að skoða of margar stöður, sérstaklega þær sem færa okkur fjær réttu lausninni.
- Til eru tvær aðferðir sem eru svipaðar fyrri leitaraðferðum: A* og IDA*.
- Til að nota þær þurfum við að geta greint stöður og gefið þeim stig eftir hversu góðar þær eru. Þetta þekkist sem brjóstvitsaðferð (e. heuristic).

Lausnir frh.

- Viljum forðast að skoða of margar stöður, sérstaklega þær sem færa okkur fjær réttu lausninni.
- Til eru tvær aðferðir sem eru svipaðar fyrri leitaraðferðum: A* og IDA*.
- Til að nota þær þurfum við að geta greint stöður og gefið þeim stig eftir hversu góðar þær eru. Þetta þekkist sem brjóstvitsaðferð (e. heuristic).
- Misgóðar greiningar í boði, sumar gefa alltaf stystu lausnina.

Lausnir frh.

- Viljum forðast að skoða of margar stöður, sérstaklega þær sem færa okkur fjær réttu lausninni.
- Til eru tvær aðferðir sem eru svipaðar fyrri leitaraðferðum: A* og IDA*.
- Til að nota þær þurfum við að geta greint stöður og gefið þeim stig eftir hversu góðar þær eru. Þetta þekkist sem brjóstvitsaðferð (e. heuristic).
- Misgóðar greiningar í boði, sumar gefa alltaf stystu lausnina.
  - Fjöldi reita í rangri staðsetningu

Rúnaheimur

Lausnir frh.

- Viljum forðast að skoða of margar stöður, sérstaklega þær sem færa okkur fjær réttu lausninni.
- Til eru tvær aðferðir sem eru svipaðar fyrri leitaraðferðum: A* og IDA*.
- Til að nota þær þurfum við að geta greint stöður og gefið þeim stig eftir hversu góðar þær eru. Þetta þekkist sem brjóstvitsaðferð (e. heuristic).
- Misgóðar greiningar í boði, sumar gefa alltaf stystu lausnina.
  - Fjöldi reita í rangri staðsetningu
  - Summa manhattan fjarlægða reitanna frá réttum stöðum

Lausnir frh.

- Viljum forðast að skoða of margar stöður, sérstaklega þær sem færa okkur fjær réttu lausninni.
- Til eru tvær aðferðir sem eru svipaðar fyrri leitaraðferðum: A* og IDA*.
- Til að nota þær þurfum við að geta greint stöður og gefið þeim stig eftir hversu góðar þær eru. Þetta þekkist sem brjóstvitsaðferð (e. heuristic).
- Misgóðar greiningar í boði, sumar gefa alltaf stystu lausnina.
  - Fjöldi reita í rangri staðsetningu
  - Summa manhattan fjarlægða reitanna frá réttum stöðum
  - og margt fleira.

Rúnaheimur

Lausnir frh.

- Viljum forðast að skoða of margar stöður, sérstaklega þær sem færa okkur fjær réttu lausninni.
- Til eru tvær aðferðir sem eru svipaðar fyrri leitaraðferðum: A* og IDA*.
- Til að nota þær þurfum við að geta greint stöður og gefið þeim stig eftir hversu góðar þær eru. Þetta þekkist sem brjóstvitsaðferð (e. heuristic).
- Misgóðar greiningar í boði, sumar gefa alltaf stystu lausnina.
  - Fjöldi reita í rangri staðsetningu
  - Summa manhattan fjarlægða reitanna frá réttum stöðum
  - og margt fleira.
- Með réttri brjóstvitsaðferð er hægt að nálægt fullri lausn.

Rúnaheimur

Full lausn

- Ef það eru tvær raðir og einn dálkur þá þurfum við í versta falli að framkvæma U.

Full lausn

- Ef það eru tvær raðir og einn dálkur þá þurfum við í versta falli að framkvæma U.
- Ef það eru bara tvær raðir þá viljum við leysa dálkinn lengst til vinstri.

Rúnaheimur

Full lausn

- Ef það eru tvær raðir og einn dálkur þá þurfum við í versta falli að framkvæma U.
- Ef það eru bara tvær raðir þá viljum við leysa dálkinn lengst til vinstri.
- Ef það eru þrjár eða fleiri raðir þá viljum við leysa efstu röðina fyrst.

Rúnaheimur

Full lausn

- Ef það eru tvær raðir og einn dálkur þá þurfum við í versta falli að framkvæma U.
- Ef það eru bara tvær raðir þá viljum við leysa dálkinn lengst til vinstri.
- Ef það eru þrjár eða fleiri raðir þá viljum við leysa efstu röðina fyrst.
- Til að leysa röð getum við leyst fyrir dálkinn lengst til vinstri, svo næst lengst til vinstri, og svo framvegis. Þurfum að leysa síðasta dálkinn í röðinni á sérstakann máta.

Rúnaheimur

Full lausn

- Ef það eru tvær raðir og einn dálkur þá þurfum við í versta falli að framkvæma U.
- Ef það eru bara tvær raðir þá viljum við leysa dálkinn lengst til vinstri.
- Ef það eru þrjár eða fleiri raðir þá viljum við leysa efstu röðina fyrst.
- Til að leysa röð getum við leyst fyrir dálkinn lengst til vinstri, svo næst lengst til vinstri, og svo framvegis. Þurfum að leysa síðasta dálkinn í röðinni á sérstakann máta.
- Þessar reglur skilgreina endurkvæma lausnaraðferð.