

RAKNINGARVENSL

summur af eftirfarandi hafa sömu útkomu, $\left(\frac{n(n+1)}{2}\right)$ sem er $O(n^2)$

$$\sum_{k=0}^n k = 0 + 1 + 2 + \dots + n = O(n^2)$$
$$\sum_{k=0}^n n - k = n + (n - 1) + (n - 2) + \dots + 0 = O(n^2)$$

summur kvótaraða hafa líka sína eigin formúlu,

$$\sum_{k=0}^n r^k = \begin{cases} r \neq 1 := \frac{r^{n+1}-1}{r} - 1 \\ r = 1 := (n + 1) \end{cases}$$

líka hægt að nota þetta stundum, þegar $0 < |r| < 1$

$$\sum_{k=0}^{\infty} r^k = \frac{1}{1 - r} \approx 1$$

LOGARITMAR

$$x^{\log_x(y)} = y \quad x^{\log_x(y)+1} = x^{\log_x(y)} \cdot x^1 = x^{\log_x(y)} \cdot x = x^{\log_x(x \times y)} = x \times y$$

ef hægt er að mynda x sem z í veldi y þá gengur eftirfarandi
$$x^{\log_z(n)} = (z^y)^{\log_z(n)} = z^{y \times \log_z(n)} = z^{\log_z(n^y)} = n^y$$

MASTER THEOREMS

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n^c)$$

- 1. Ef $c < \log_b(a)$ þá er $T(n) = O(n^{\log_b(a)})$
- 2. Ef $c = \log_b(a)$ þá er $T(n) = O(n^{\log_b(a)} \cdot \log(n))$
- 3. Ef $c > \log_b(a)$ þá er tími reikniritisins $T(n) = O(f(n))$

$$T(n) = a \cdot T(n - b) + f(n)$$

- 1. Ef $a < 1$ þá er $T(n) = O(f(n))$
- 2. Ef $a = 1$ þá er $T(n) = O(n \cdot f(n))$
- 3. Ef $a > 1$ þá er $T(n) = O(f(n) \cdot n^{\frac{a}{b}})$

LEGGIR OG LEIÐINDI

leggir skilgreindir sem $U \rightarrow V$ þar sem U er upphafspunktur og V er endapunktur.

fram / trjáleggir

V er nýr þegar djúpleit á U hefst:

$$U.pre < V.pre < V.post < U.post$$

- 1. **trjáleggur** ef U er foreldri V
- 2. **framleggur** annars

bakleggur

v er virkur þegar dýptarleit á U hefst:

$$V.pre < U.pre < U.post < V.post$$

krossleggur

V búinn lokið þegar dýptarleit á U hefst: $V.post < U.pre$

DYNAMIC PROGRAMMING

við höfum runu spilapeninga t_1, \dots, t_n . hver spilapeningur hefur tvo eiginleika, virði og lit. virði er tala á forminu $2, 4, 8, \dots, 2048$ og litir eru R, G eða B . Við getum myndað vensl á milli peninga ef þeir uppfylla eftirfarandi reglur:

- þeir hafa virði, eða
- þeir eru í sama lit, eða
- annar þeirra hefur virði 2

við viljum finna lengstu hlutrunu spilapeninga $t_{i_1} \dots t_{i_k}$ þar sem $i_1 < \dots < i_k$.

Tökum sem dæmi rununa $4R, 4B, 16R, 16G, 2R$

skref

- 1. lýsum undirverkefnum
 - undirverkefni hér væri að finna lengstu venslu hlutrunu frá hverju staki
- 2. lýsum ákvörðun / ágiskun í hverju undirskrefi
 - hér þyrftum við að ákvarða hvaða stak væri best að halda áfram með í rununninni
 - það er frekar augljóst að besta stakið til bæta núverandi runu er það sem hefur lengstu hlutrunu nú þegar
- 3. lýsum rakningarvenslum
 - tökum skrefin á undan og bætum við grunntilviki

$$f(i) = \begin{cases} i = n : 1 \\ i \neq n : \max(arr \in t_i \sim t_j \wedge i < j) \end{cases}$$

- 4. lýsum hvernig koma má í veg fyrir endurtekt með því að geyma niðurstöðurstöður
 - hérna þurfum við að vita hvað besta runa af stökum $A[i + 1 \dots]$
 - við geymum þær upplýsingar í einföldu fylki þar sem hvert stak táknar lengstu runu frá því staki
- 5. rökstyðjum tímaflækju
 - í hverju skrefi erum við að velja $max(arr)$ og sú aðferð ítrar yfir öll stök fylkisins í $O(n)$ tíma
 - við endurtökum þessi undirskref fyrir öll stök fylkisins
 - lokatímaflækja er $O(n * n) = O(n^2)$

NETAREIKNIRIT

Net G=(V,E)	Vogtölur	Reiknirit	Tímaflækja
ó/stefnt, rásað	$w \in \mathbb{R}, w \geq 0$	Dijkstra	$O(\log(V)E)$
ó/stefnt, rásað	$w \in \mathbb{R}$	Bellman-Ford	$O(VE)$
stefnt, órásað	$w \in \mathbb{R}$	SSSPDAG	$O(V + E)$

```
def dijkstra(G, start):
    dist, parent = {}, {}
    for v in G.v: dist[v]
    Q = new minPQ(start)

    while Q:
        v = Q.pop()
        for u in G[v]:
            if dist[u] > dist[v] + G[v][u]:
                dist[u] = dist[v] + G[v][u]
                parent[u] = v
            Q.push(u)
```

MaxFlow ($G = (V,E)$) O(VE)

Tekur inn flæðisnet og skilar því með hámarksflæði, passa að leggir þurfa þyngd, _____

FlowToPaths (MaxFlow) O(E)

Tekur inn flæðisnet, býið að finna maxflow og skilar vegum sem fylgja því með tilliti til hnúta, _____

FlowToMatching (MaxFlow) O(E)

Tekur inn maxflow og varpar yfir í spyrðingu, _____

Maximum Matching (G(V,E))

Tekur inn net með hnúta af típu inn-út, og skilar hámarksspyrðingu, líka hægt að fá með MaxFlow, _____

Matching To Cover (Matching)

Tekur inn spyrðingu úr falli eins og MM og skilar þakningu yfir netið, _____

LÍNULEG BESTUN

Formúla línu er $y = a \cdot x + b$ þar sem a er hallatala línu og b er skurðpunktur við y ás. Til að finna skurðpunkt lína setja upp jöfnuhneppi og leysa fyrir x .

Fjöldi skurðpunkta útfra skorðum er $\binom{n}{2} \rightarrow nCr$ þar sem n er fjöldi skorða, á meðan hornpunktar gjaldgenga svæðisins tákna bara innliggjandi horn, sést mjög auðveldlega á mynd.

P/NP

verkefni sem hægt er að leysa í margliðutíma eru í flokknum P, verkefni sem ekki hægt er að leysa í margliðutíma eru í flokknum NP.

- **Ákvörðunarverkefni:** verkefni sem hafa lausn já/nei
 - **P:** hægt að leysa í margliðutíma
 - **NP:** hægt að staðfesta já á margliðutíma
 - líka ef hægt er að leysa þekkt NP-verkefni með lausn á þessu verkefni
 - **co-NP:** hægt að staðfesta nei á margliðutíma

Reynum að finna minnsta *sterka* mengi hnúta í neti G . Setjum verkefnið fram sem ákvörðunarverkefni, þ.e. svörum fyrir gefna tölu k hvort til sé sterkt mengi af stærð k í netinu. Við getum ekki svarað því í margliðutíma en við getum, ef við fáum gefið mengi þá getum við svarað í margliðutíma hvort það sé af stærð k eða ekki. Þetta er NP-verkefni (*co-NP*).

SLEMBIREIKNIRIT

Líkur á atburði A eru táknaðar með $\Pr[A]$ og fengnar með $\sum_{w \in A} \Pr[w]$ þ.e. fyrir tening með fjórar hliðar er mengi sléttra talna $\Omega = \{2, 4\}$ og líkurnar á að fá aðra þeirra eru $\frac{1}{4} + \frac{1}{4} = \frac{2}{4}$

Fyrir tvo fjögurra hliða teninga eru heildafjöldi útkoma hjá okkur 4^2 þannig mengi þar sem báðir teningar hafi slétta tölu er $\Omega = \{(2, 2), (4, 2), (2, 4), (4, 4)\}$ og líkurnar þá $4 * \frac{1}{16} = \frac{4}{16} = \frac{1}{4}$.

Þetta virkar þar sem fyrir sérhverja tvo atburði A og B með $\Pr[B] > 0$ skilgreinum við skilyrtar líkur á A gefið B , þ.e. A og B , sem $\Pr[A \mid B] = \frac{\Pr[A \wedge B]}{\Pr[B]}$

Líkur á að fá í mesta lagi einn 3 þegar við köstum tveimur tengingum og vitum að fyrri teningurinn skilar alltaf 3 eru $\frac{3}{4}$, sjáum að ef við notum formúluna fyrir ofan er A að fá ekki þrist á öðrum teningnum, B er að fyrri teningur skilar alltaf 3. $\Pr[A] = \frac{3}{4}$ og $\Pr[B] = \frac{1}{4}$. Þessu er síðan hægt að plugga inn í formúluna uppi.

Væntigildi $\max(X_1, X_2)$, þar sem X_1 og X_2 eru fjögurra hliða teningar, höfum við útkomumengi $\Omega = \{(1, 1), (1, 2), (1, 3), \dots, (4, 3), (4, 4)\}$

Væntigildið er þá summa líkna þess að fá gildi, margfaldað við gildi þ.e. $1 * \frac{1}{16} + 2 * \frac{3}{16} + 3 * \frac{5}{16} + 4 * \frac{7}{16}$

Þetta er svo mikið rugl??

TIPS N TRICKS

- Kvik bestunardæmi sem hægt er að leysa með þríhyrnings geymslu, *tvívítt fylki sem geymir "bestu" lausn fyrir þann reit*, eru yfirleitt $M \cdot N$
- Línuleg bestun er auðveld, bara plugga skorðum inn í simplex
- Hafa gaman :)
- Drekkja jafnt magn vatns og áfengis á djamminu
- Alltaf hafa hreinar nærbuxur aðgengilegar

Vaxtarhraði

$1 < \log(n) < (\log(n))^c < \sqrt{n} < n < n \log(n) < n^c < c^n < n^n < n!$
Þar sem c er einhver fasti