

2.3 Keskimääräisen tapauksen analyysi

Muistetaan

$$T_{\text{ave}}(n) = \sum_{|x|=n} P_n(x)T(x)$$

missä $|x|$ on tapauksen x koko ja P_n jakauma kokoa n oleville tapauksille.

Siis $T_{\text{ave}}(n)$ on satunnaismuuttujan $T(x)$ odotusarvo jakauman P_n suhteen; merkitsemme $T_{\text{ave}}(n) = E_{P_n}[T(x)]$ tai pelkästään $T_{\text{ave}}(n) = E[T(x)]$ jos jakauma on selvä asiayhteydestä.

Odotusarvon perusominaisuuksia (esim. TN I):

- $E[X] + E[Y] = E[X + Y]$ ja $E[aX] = aE[X]$ aina (lineaarisuus)
- $E[XY] = E[X]E[Y]$ jos X ja Y ovat riippumattomia (mitä merkitään $X \perp Y$)

Esimerkki Peräkkäishaku: löydettävä alkion x indeksi taulukossa $A[1 \dots n]$ (jos löytyy)

```
search( $A[1 \dots n], x$ ):  
1.    $i := 1$   $\Theta(1)$   
2.   while  $i < n$  and  $A[i] \neq x$  do  $k(x) \cdot \Theta(1)$   
3.        $i := i + 1$ ;  
4.   if  $A[i] = x$  then return  $i$   $\Theta(1)$   
       else return "ei löydy"
```

Aikavaatimuksessa merkitään

$k(x)$ = rivin 3 suorituskertojen lkm. syötteellä x .

Selvästi $T(x) = ak(x) + \Theta(1)$ joillain $a, b > 0$.

Jakaumaoletukset:

- taulukon A alkiot aina erisuuria
- taulukon A alkioiden kaikki järjestykset yhtä todennäköisiä
- alkio x on taulukossa A todennäköisyydellä q

Olkoon X_i niiden tapausten (A, x) joukko, joilla $x = A[i]$.

Jakaumaoletuksen mukaan

$$P_n(X_i) = P_n(X_j) \quad \text{kaikilla } 1 \leq i, j \leq n$$

$$\sum_{i=1}^n P_n(X_i) = q$$

joten $P(X_i) = q/n$ kaikilla i .

Koska

$$k(x) = \begin{cases} i - 1 & \text{jos } x = A[i] \\ n - 1 & \text{jos } x \neq A[i] \end{cases} \text{ kaikilla } i,$$

saadaan

$$\begin{aligned} k_{\text{ave}}(n) &= \sum_{i=1}^n (i - 1)P_n(X_i) + (n - 1)\left(1 - \sum_{i=1}^n P_n(X_i)\right) \\ &= \sum_{i=0}^{n-1} i \frac{q}{n} + (n - 1)(1 - q) \\ &= \frac{q n(n - 1)}{n \cdot 2} + (n - 1)(1 - q) \\ &= \left(1 - \frac{q}{2}\right)n + \frac{q}{2} - 1. \end{aligned}$$

Siis

$$T_{\text{ave}}(n) = a k_{\text{ave}}(n) + \Theta(1) = a\left(1 - \frac{q}{2}\right)n + \Theta(1).$$

Jos $q = 1$ tutkitaan keskimäärin puolet taulukosta; jos $q = 1/2$ tutkitaan 3/4.

Sanakirjaongelma

Ongelma on sinänsä yksinkertainen, mutta antaa mahdollisuuden esitellä joitakin keskimääräisen tapauksen analyysin (ja myöhemmin tasoitetun analyysin) tekniikoita.

On annettu äärellinen joukko **avaimia** $A = \{a_1, \dots, a_n\}$. Tehtävänä on ylläpitää joukkoa $S \subseteq A$, kun seuraavat operaatiot ovat sallittuja:

access(i): palauttaa **true** jos $a_i \in S$, muuten **false**

insert(i): $S := S \cup \{a_i\}$

delete(i): $S := S - \{a_i\}$

Huomautuksia:

- Käytännössä idea on yleensä, että kuhunkin avaimeen a_i liittyy jokin data x_i , jonka insert tallettaa ja access palauttaa. Yksinkertaisuuden vuoksi esitetään tässä vain perusversio.
- yksinkertaisuuden vuoksi valitaan $a_i = i$ (siis $A = \{1, \dots, n\}$).
- tehokkaita ratkaisumenetelmiä: hajautus, hakupuut
- seuraavassa analysoidaan linkitettyyn listaan perustuvia yksinkertaisia ratkaisuja

Perustoteutus linkitettyllä listalla:

access(i): käydään listaa järjestyksessä läpi kunnes i löytyy tai päästään loppuun

insert(i): jos i ei listassa, lisätään se loppuun

delete(i): jos i listassa, poistetaan; muuten ei muutoksia

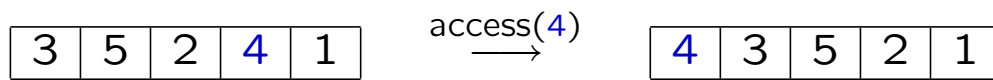
Seuraavassa tarkastellaan edistyneempiä versioita, joissa access-operaation yhteydessä listaa mahdollisesti järjestellään uudelleen jonkin heuristiikan mukaan.

Olkoon $L[k]$ talletusrakenteena olevan listan k :s alkio, $k = 1, \dots, |S|$.

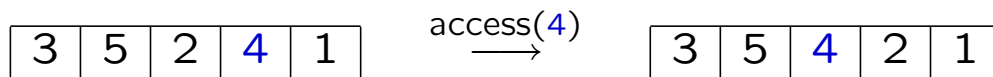
- jos $i = L[k]$ niin alkioon i kohdistuvat operaatiot vaativat k vertailua " $L[j] = i?$ "
 - vertailujen laskeminen antaa selvästi oikean kertaluokan operaatioiden koko suoritusajalle
- ⇒ pyritään saamaan usein haettavat alkiot listan alkupäähän

Tarkastelemme jatkossa seuraavia heuristiikkoja:

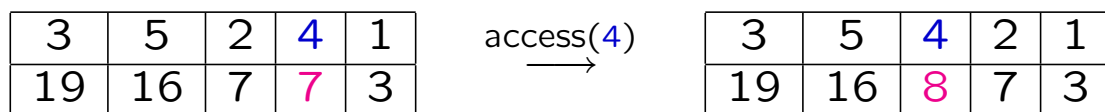
Move-to-Front (lyh. MF): haettu alkio siirretään listan keulaan



Transpose (lyh. TR): haettua alkioita siirretään yksi askel kohti listan keulaa



Frequency Count (lyh. FC): pidetään kustakin alkioista yllä laskuria siihen kohdistuneista operaatioista; pidetään lista laskurien mukaan laskevassa järjestyksessä



(kuvassa viitelaskurit alarivissä)

insert-operaatiossa alkio lisätään alustavasti listan häntään ja sitten kohdistetaan siihen yksi ”ylimääräinen” access-operaatio (siis oikeasti MF lisää listan keulaan, TR toiseksi viimeiseksi jne.)

Keskimääräisen tapauksen analyysia varten tehdään seuraavat oletukset suoritettavien operaatioiden jakaumasta:

- insert- ja delete-operaatioita ei tule, joukko sisältää jatkuvasti tasan alkioita $1, \dots, n$
- hetkellä t valitaan suoritettavaksi operaatio $\text{insert}(i)$ todennäköisyydellä p_i , kun $i = 1, \dots, n$
- eri ajanhetkillä valittavat operaatiot ovat toisistaan riippumattomia

Tässä siis $p_i \geq 0$ ja $\sum_{i=1}^n p_i = 1$. Yksinkertaisuuden vuoksi oletetaan avaimet nimetyin niin, että $p_1 \geq p_2 \geq \dots \geq p_n$.

Kun (p_1, \dots, p_n) on annettu, jakaumaoletuksen vallitessa voidaan vielä määritellä seuraava "heuristiikka":

Decreasing Probability (lyh. DP): pidä lista kiinteässä todennäköisyyden mukaan laskevassa järjestyksessä: $L[i] = i$ kaikilla i

Analyysin tausta-ajatus on nyt seuraava:

- DP on **offline**-heuristiikka: se vaatii todennäköisyyksien tietämisen ennen kuin toiminta voi alkaa
- DP on (kuten kohta nähdään) optimaalinen jos todennäköisyydet tunnetaan
- MF, TR ja FC ovat **online**-heuristiikkoja: ne mukautuvat samalla kun toiminta etenee
- jos jakaumaoletus pätee mutta todennäköisyyksiä ei tunneta etukäteen, joudutaan käyttämään jotakin online-heuristiikkaa
- halutaan osoittaa, että millä tahansa (p_1, \dots, p_n) esim. MF (joka ei "tiedä" näitä todennäköisyyksiä) on melkein yhtä tehokas kuin DP (joka on optimoitu juuri näille todennäköisyyksille)

(Seuraava keskimääräisen tapauksen esitys perustuu artikkeliin Rivest: On self-organizing sequential search heuristics, CACM 1985.)

Sanakirjaongelma muistuttaa läheisesti virtuaalimuistin sivutusongelmaa:

- ajatellaan kaikki virtuaalimuistin sivut järjestetyksi listaan
- keskusmuistissa pidetään K ensimmäistä sivua listalta, K keskusmuistin koko
- esim. MF-listanjärjestysheuristiikka vastaa LRU-sivutusmenetelmää (Least Recently Used)
- sivutusongelmassa kuitenkin kustannusfunktio on monimutkaisempi: jos $L[i]$ on sivun i sijainti listassa, niin sivuun i kohdistuvan operaation kustannus on 0 jos $L[i] \leq K$ ja 1 muuten (lasketaan siis sivunpuutoksia)

Erityisesti tässä sovelluksessa aiemmin esitetty jakaumaoletus ei ole realistinen, joten [tasoitettu analyysi](#) on järkevämpää kuin keskimääräisen tapauksen analyysi. (Teemme jatkossa myös tasoitetun analyysin sanakirjaongelmalle.)

Esitellään tarvittavat merkinnät päätulosten
formuloimiseksi:

$$\begin{aligned}(i_1, \dots, i_m) &= \text{operaatiojono } \text{access}(i_1), \dots, \text{access}(i_m) \\ P_m(x) &= \text{pituudeltaan } m \text{ olevan} \\ &\quad \text{operaatiojonon } x \text{ todennäköisyys}\end{aligned}$$

Siis

$$P_m(i_1, \dots, i_m) = p_{i_1} p_{i_2} \dots p_{i_m}.$$

Kun A on jokin em. algoritmeista
($A \in \{MF, TR, FC, DP\}$) merkitsemme

$$\begin{aligned}T^A(x) &= \text{operaatiojonon } x \text{ vaatimien vertailujen lkm.} \\ T_{\text{ave}}^A(m) &= m \text{ operaation keskimäär. vertailujen lkm.} \\ &= \sum_{|x|=m} P_m(x) T^A(x).\end{aligned}$$

Seuraavassa analysoimme eri algoritmien
asymptoottista keskimääräistä kustannusta

$$T_{\text{ave}}^A = \lim_{m \rightarrow \infty} \frac{1}{m} T_{\text{ave}}^A(m).$$

Ilmeisesti (sopivalla toteutuksella) operaatiojonon x
koko suoritusaika on muotoa $aT^A(x) + \Theta(1)$ missä
vakio a on sama kaikille em. algoritmeille.
Tulkitsemme siis jatkossa suoraan että T^A on
algoritmin A suoritusaika.

Tavoitteena on todistaa seuraavat tulokset:

1. DP on optimaalinen: $T_{\text{ave}}^{\text{DP}}(m) \leq T_{\text{ave}}^A(m)$ mille tahansa A (muillekin kuin em. heuristiikoille) ja kaikille m
2. asympotoottisesti myös FC on optimaalinen:
 $T_{\text{ave}}^{\text{FC}} = T_{\text{ave}}^{\text{DP}}$.
3. MF vie korkeintaan kaksi kertaa niin paljon aikaa kuin DP: $T_{\text{ave}}^{\text{MF}} \leq 2T_{\text{ave}}^{\text{DP}}$.
4. TR on ainakin yhtä hyvä kuin MF: $T_{\text{ave}}^{\text{TR}} \leq T_{\text{ave}}^{\text{MF}}$ (ja epäyhtälö on ei-triviaaleissa tapauksissa aito)

Todistukset perustuvat seuraavanlaisiin tekniikoihin:

1. odotusarvon perusominaisuudet
2. suurten lukujen laki
3. suoraviivainen lasku
4. Markovin ketjut

Eryteisesti kohdat (1) ja (2) tehdään harjoituksen vuoksi melko yksityiskohtaisesti.

Kohta (4) on tärkeä uusi tekniikka.

Kun π on joukon $\{1, \dots, n\}$ permutaatio (siis bijektio $\{1, \dots, n\} \rightarrow \{1, \dots, n\}$), sanotaan että lista L on järjestyksessä π jos $L[\pi(i)] = i$ kaikilla i . Jos lista on järjestyksessä π , niin operaation $\text{access}(i)$ kustannus on $\pi(i)$, joten keskimääräinen kustannus on

$$\sum_{i=1}^n p_i \pi(i).$$

Olkoon $P_t^A(\pi)$ todennäköisyys että ajanhetkellä t (eli ensimmäisten $t - 1$ operaation jälkeen) algoritmin A lista on järjestyksessä π .

Erityisesti DP pitää listansa vakiojärjestyksessä:

$$P_t^{\text{DP}}(\pi) = \begin{cases} 1 & \text{jos } \pi(i) = i \text{ kaikilla } i \\ 0 & \text{muuten} \end{cases} \quad \text{kaikilla } t.$$

Merkitään tätä vakiojärjestystä π_{opt} .

Olkoon $S_{\text{ave}}^A(t)$ algoritmin A operaation numero t keskimääräinen kustannus (siis vertailuina mittattuna). Siis

$$S_{\text{ave}}^A(t) = \sum_{\pi} P_t^A(\pi) \sum_{i=1}^n p_i \pi(i)$$

ja erityisesti

$$S_{\text{ave}}^{\text{DP}}(t) = \sum_{i=1}^n i p_i \quad \text{kaikilla } t.$$

Olkoon π järjestys jossa alkio i on ennen alkioita j , eli $\pi(i) < \pi(j)$. Jos nyt $p_i < p_j$, niin

$$p_i\pi(j) + p_j\pi(i) < p_i\pi(i) + p_j\pi(j).$$

Siis jos edelleen π' on muuten sama kuin π paitsi että $\pi'(i) = \pi(j)$ ja $\pi'(j) = \pi(i)$, niin pätee

$$\sum_{k=1}^n p_k\pi(k) > \sum_{k=1}^n p_k\pi'(k).$$

Toistamalla tätä argumenttia nähdään, että $\sum_{k=1}^n p_k\pi(k)$ saa pienimmän arvonsa kun järjestys π on alkioiden todennäköisyyksien mukaan laskeva, eli π_{opt} .

Siis kaikilla π pätee

$$\sum_{k=1}^n p_k\pi(k) \geq \sum_{k=1}^n p_k\pi_{\text{opt}}(k) = \sum_{k=1}^n kp_k = S_{\text{ave}}^{\text{DP}}(t)$$

joten millä tahansa algoritmilla A pätee

$$\begin{aligned} S_{\text{ave}}^A(t) &= \sum_{\pi} P_t^A(\pi) \sum_{i=1}^n p_i\pi(i) \\ &\geq \sum_{\pi} P_t^A(\pi) S_{\text{ave}}^{\text{DP}}(t) \\ &= S_{\text{ave}}^{\text{DP}}(t). \end{aligned}$$

Odotusarvon lineaarisuudesta seuraa

$$T_{\text{ave}}^A(m) = \sum_{t=1}^m S_{\text{ave}}^A(t)$$

joten edellisen perusteella saadaan

Lause Kaikilla A ja m pätee

$$T_{\text{ave}}^{\text{DP}}(m) \leq T_{\text{ave}}^A(m).$$

□

Koska edelleen

$$T_{\text{ave}}^A = \lim_{m \rightarrow \infty} \frac{1}{m} \sum_{t=1}^m S_{\text{ave}}^A(t),$$

nähdään helposti että

$$T_{\text{ave}}^A = \lim_{t \rightarrow \infty} S_{\text{ave}}^A(t)$$

mikäli tämä raja-arvo on olemassa. Erityisesti tapauksessa $A = \text{DP}$ kustannus $S_{\text{ave}}^A(t)$ ei riipu ajanhetkestä t ja saadaan

Lause

$$T_{\text{ave}}^{\text{DP}} = \sum_{i=1}^n ip_i.$$

□

Todistetaan seuraavaksi, että asympotoottisesti FC on yhtä hyvä kuin DP.

Intuitiivisesti päättely on seuraava:

1. FC on muuten sama kuin DP, paitsi että todennäköisyyksiä p_i approksimoidaan suhteellisilla frekvensseillä \hat{p}_i
2. suurten lukujen lain nojalla $\hat{p}_i \rightarrow p_i$ todennäköisyydellä 1
3. siis todennäköisyydellä 1 jostain ajanhetkestä alkaen $\hat{p}_i < \hat{p}_j$ jos ja vain jos $p_i < p_j$
4. siis todennäköisyydellä 1 jostain ajanhetkestä alkaen algoritmien FC ja DP listat ovat samassa järjestyksessä (mahdollisesti lukuunottamatta pareja (i, j) joilla $p_i = p_j$)
5. siis rajalla $t \rightarrow \infty$ algoritmit FC ja DP käyttäytyvät samalla tavalla

Muodollisempi todistus edellyttää puhumista äärettömän pitkistä operaatiojonoista.

Kun x on äärettömän pitkä jono access-operaatioita, olkoon x^m sen ensimmäiset m operaatiota käsittävä osajono.

Olkoon P jakaumaoletuksen mukainen jakauma äärettömän pitkille jonoille, siis

$$P(\{x \mid x^m = (x_1, \dots, x_m)\}) = P_m(x^m) = p_{i_1} \dots p_{i_m}.$$

Olkoon $\hat{p}_i(x^m)$ operaation access(i) suhteellinen frekvenssi operaatiojonossa x^m .

Lemma Jos x valitaan jakauman P mukaan, niin todennäköisyydellä 1 pätee

$$\lim_{m \rightarrow \infty} \hat{p}_i(x^m) = p_i$$

kaikilla i .

Todistus Seuraa suoraan vahvasta suurten lukujen laista; ks. todennäköisyyslaskennan oppikirjat. \square

Toinen tarvittava aputuloks on

Rajoitetun konvergenssin lause Oletetaan, että

- f_1, f_2, f_3, \dots on jono tasaisesti rajoitettuja satunnaismuuttujia, ts. jollain M pätee

$$|f_n(x)| \leq M \quad \text{kaikilla } n, x,$$

- $E[f_n]$ on olemassa kaikilla n ja
- $f_n \rightarrow f$ melkein kaikkialla, ts.

$$P(\{x \mid \lim_{n \rightarrow \infty} f_n(x) = f(x)\}) = 1.$$

Nyt

$$E[f] = \lim_{n \rightarrow \infty} E[f_n].$$

□

Tämä kertoo sen intuitiivisesti uskottavan seikan, että raja-arvon ja odotusarvon voi ottaa kummassa järjestyksessä tahansa.

Tämä ei kuitenkaan ole triviaalia kun puhutaan äärettömistä joukoista, joten on syytä todella tarkistaa lauseen ehdot.

Mittateorian (integraalilaskennan, todennäköisyyslaskennan) oppikirjoista löytyy lisää vastaavia lauseita, joissa ehdot ovat hieman toiset.

Lause

$$T_{\text{ave}}^{\text{FC}} = T_{\text{ave}}^{\text{DP}}.$$

Todistus Edellisen mukaan kun x valitaan jakauman P mukaan, pätee todennäköisyydellä 1

$$\lim_{m \rightarrow \infty} \hat{p}_i(x^m) = p_i \quad \text{kaikilla } i.$$

Eryteisesti todennäköisyydellä 1 on olemassa sellainen m_0 , että kun $m \geq m_0$ niin pätee

$$\hat{p}_i(x^m) > \hat{p}_j(x^m) \quad \text{aina kun } p_i > p_j.$$

Olkoon $M(x) = m_0$ jos tällainen m_0 on olemassa, muuten $M(x) = \infty$. Siis ajanhetkestä m_0 eteenpäin algoritmien FC ja DP listat ovat samassa järjestyksessä lukuunottamatta mahdollisesti sellaisia alkiopareja joiden todennäköisyydet ovat samat.

Olkoon $S_{\text{ave}}^A(x^m)$ operaation numero $m + 1$ keskimääräinen kustannus algorithmillä A , kun edeltävät operaatiot ovat jonon x^m mukaiset.

Siis aina pätee

$$S_{\text{ave}}^{\text{DP}}(x^m) = T_{\text{ave}}^{\text{DP}}$$

ja lisäksi kun $m \geq M(x)$ pätee

$$S_{\text{ave}}^{\text{FC}}(x^m) = S_{\text{ave}}^{\text{DP}}(x^m)$$

joten todennäköisyydellä 1

$$\lim_{m \rightarrow \infty} S_{\text{ave}}^{\text{FC}}(x^m) = T_{\text{ave}}^{\text{DP}}.$$

Selvästi $E[S_{\text{ave}}^{\text{FC}}(x^{m-1})]$ on olemassa kaikilla m , ja triviaalisti aina $S_{\text{ave}}^{\text{FC}}(x^{m-1}) \leq n$. Rajoitetun konvergenssin lauseen nojalla nyt

$$\lim_{m \rightarrow \infty} E[S_{\text{ave}}^{\text{FC}}(x^{m-1})] = E[T_{\text{ave}}^{\text{DP}}].$$

Koska toisaalta

$$\begin{aligned} T_{\text{ave}}^{\text{FC}} &= \lim_{m \rightarrow \infty} S_{\text{ave}}^{\text{FC}}(m) \\ &= \lim_{m \rightarrow \infty} E[S_{\text{ave}}^{\text{FC}}(x^{m-1})] \end{aligned}$$

ja toisaalta triviaalisti

$$E[T_{\text{ave}}^{\text{DP}}] = T_{\text{ave}}^{\text{DP}},$$

väite seuraa. \square

Tietojenkäsittelytieteellisissä artikkeleissa sellaiset tekniset apuvälineet kuin suurten lukujen laki ja rajoitetun konvergenssin lause sivuutetaan usein maininnalla (tai ilman mainintaa).

Tässä on esimerkin vuoksi asia esitetty melko yksityiskohtaisesti, koska näistä asioista on syytä kuitenkin olla tietoinen (ja todennäköisyyslaskennassa voi olla vaarallista luottaa intuitioon).