الگوریتم
رتبه

محسن هوشمند
دانشکده علم رایانه و تکنولوژی اطلاعات
دانشگاه تحصیلات تکمیلی علوم پایه زنجان

رتبه

حجم زیاد علیه درک مناسب

- سختی فهم انسان از استخراج اطلاعات در حجم زیاد داده‌های بدون ساختار
- تلخیص داده‌ها با استخراج کمیت‌های آماری
  - از راه‌حل‌های مهم در مواجهه با چنین داده‌هایی

الگوریتم‌ها و ساختارهای داده محاسبه ویژگی‌های مبتنی بر رتبه داده‌ها

- با استفاده از حافظه کم و یک گذر از داده‌ها

چندک

پرکاربردترین نوع رتبه
- چندک‌ها

q - چندک
- $(0 \leq q \leq 1)$
- مقداری از دنباله مقادیر
  - کسر q از مقادیر دنباله کمتر یا مساوی آن
  - 1 - q باقی‌مانده بیشتر یا مساوی آن

در صورت دنباله دارای n مقدار
- مقدار q–چندک مقداری از دنباله با رتبهٔ $q \cdot n$

صدک‌ها یا درصدها

چندک‌های تقسیم‌کنندهٔ دنباله مرتب شده به ۱۰۰ قسمت مساوی

95-اُمین صدک متناظر ۰٫۹۵–چندک

چندک‌های ۰ و ۱ به ترتیب حداقل و حداکثر مقادیر در دنباله

چندک ۰٫۵ متناظر میانه مقادیر

چندک

ایان مونرو و مایکل پاترسون در سال ۱۳۵۹

- اثبات یافتن چندک خاص در دقیقا p گذر از داده‌ها نیازمند به حافظه $O(n^{\frac{1}{p}})$
- به معنای نبودن الگوریتم تک‌گذری تولید مقدار دقیق چندک در فضای زیرخطی
- تشویق به جستجوی الگوریتم‌های محاسبهٔ چندک‌های تقریبی

چندک

در عمل، قابل تحمل بودن وجود خطا در محاسبهٔ چندک

- اعمال چنین تخمین‌هائی روی داده‌های نوفه‌دار
- تقریب توزیع‌های داده ناشناخته

در نتیجه: در پی چندک ϵ-تقریبی q در بیشتر موارد

به معنای قرار گرفتن رتبهٔ مقداری $[(q - \epsilon) \cdot n, (q + \epsilon) \cdot n]$

- n تعداد مقادیر و $0 < \epsilon < 1$ پارامتر خطا
- امکان واجد شرایط بودن بیش از یک مقدار

چندک

تخمین ویژگی‌های مختلف رتبه

- مانند خلاصه‌های چندک

نقشی مهم در روش‌های تشخیص نقاط پرت جریانی

مثال

- نظارت بر تراکنش‌های تجارت الکترونیک آنلاین جهت تشخیص تقلب در کارت اعتباری
- در پی موقعیت‌ها و مکان‌های پرداخت غیرمعمول
- مقادیری خارج از صدک نود و نهم توزیع مکان معمول برای مشتریان

مثال- تشخیص تقلب

تقلب مالی از مسائل اساسی پیش روی صنعت مالی

در سال ۱۳۹۴، تقلب در کارت‌های اعتباری و نقدی جهانی منجر به زیان‌هایی بالغ بر ۲۱,۸۴ میلیارد دلار

تولید برنامه‌های جستجو و شناسایی علائم تقلب مالی

- غالبا استفاده از متغیرهای خاص متعددی
  - بررسی «درجه دورافتادگی» یا درجه پرتی آنها در هر مشاهده
- مثال، استفاده از متغیرهایی مانند کل مبلغ هزینه شده در کارت اعتباری و مقدار هزینه شده در روز

برای هر مشاهده،

- امکان تقریب درجه دورافتادگی با چندک‌های برخی از توزیع‌های هزینه
- شناسائی و برچسب زدن مشاهدات به مشکوک به تقلب با نقاط پرت
  - از طریق مقایسه با برخی چندک‌های بالا، مثل چندک ۰,۹۵

چندک

نظارت بر ترافیک وب از دیگر حوزه کاربردی عظیم خلاصه‌های رتبه

بررسی خلاصه‌ها متناظر با تشخیص زودتر مشکلات

- بدون بررسی داده‌های واقعی

مثال- نظارت بر تارمانه

مدیریت میلیون‌ها کاربر در تارمانه‌های بزرگ به صورت روزانه

شهریور و مهر ۱۳۹۶

- ویکی‌پدیا: پردازش حدود ۵۰۰ میلیون بازدید در روز در تمام زبان‌های خود
- معادل تقریباً ۵٫۷ هزار درخواست در ثانیه با استفاده از بیش از ۳۰۰ سرور در سراسر جهان

تأخیر در تارمانه یا به زبان دقیق‌تر تأخیر بین لحظه ایجاد محتوا و لحظه انتقال به بازدیدکننده

- از جمله گزینه‌های مهم عملکرد هر تارمانه
- شایع بودن چولگی در توزیع احتمالی میزان تأخیر
- اعمال نظارت معمولاً با ردیابی برخی چندک‌ها یا صدک‌های خاص بالا

رایج‌ترین سوالات

- تأخیر ۹۵ درصد از درخواست‌ها برای سرور وب منفرد چقدر است؟
- تأخیر ۹۹ درصد از درخواست‌ها برای کل وب‌سایت چقدر است؟
- تأخیر ۹۵ درصد از درخواست‌ها برای کل وب‌سایت در ۱۵ دقیقه گذشته چقدر بود؟

مثال- نظارت بر تارمانه

امکان پاسخ به چنین سوالاتی با محاسبه چندک

- دارای تفاوت از نظر فنی
- در نتیجه نیاز به استفاده از روش‌های مختلف

سوال اول

- محاسبهٔ خلاصه برای هر جریان منفرد

سوال دوم

- نیازمند الگوریتم‌های توزیع شده جهت محاسبهٔ آمار روی داده‌های جریان‌های متعدد

سوال سوم

- نیازمند به زیرمجموعه‌ای از داده‌های جریان تعریف شده در یک پنجره زمانی
- تغییر مداوم چنین زیرمجموعه‌هائی

پرسش یا پرس‌وجوی چندک

وظیفه یافتن q-چندک،

یا یافتن مقادیری از دنبالهٔ مرتب از n مقدار دارای رتبهٔ $q.n$

- $q \in (0,1)$

پرس‌وجوی میانه

- موردی خاص از پرس‌وجوی چندک با $q = 0.5$

محاسبه چندک

متاخر نبودن مسئلهٔ محاسبه چندک

- دارای روش‌های مناسب در محاسبات کلاسیک

وجود چالش‌های جدیدی برای جریان‌های نامحدود

- رایج در برنامه‌های داده بزرگ
- با داشتن محدویت در حافظه و امکان‌پذیری تک‌گذاری از داده‌ها

الگوریتم طرح کلی شمارش-کمینه

- امکان محاسبه مقادیر تقریبی چندک‌ها
- اما نیازمند حافظه بسیار بیشتر نسبت به الگوریتم‌های مورد بحث

پرس‌وجوی چندک معکوس

نوع دیگری از مسئله مربوط به چندک

یافتن رتبهٔ مقدار داده شده در دنباله‌ای مرتب از n مقدار

در صورت یافتن آن و داشتن تعداد کل مقادیر n، محاسبه چندک متناظر q:

$$q = \frac{1}{n} \cdot rank(x)$$

چندک بازه

در بسیاری از برنامه‌ها،

- اهمیت یافتن تعداد مقادیر از دنبالهٔ مرتب از n مقدار در بازهٔ [a, b]
- غالبا شناخته شده به عنوان پرس‌وجوی بازه
- محاسبه چنین عددی
  - تعیین رتبه‌های مرزهای محدوده و برگرداندن تفاوت آنها

چند روش

الگوریتم‌های نمونه‌برداری تصادفی

الگوریتم q-digest یا تلخیص-q بر مبنای درخت ساده

الگوریتم t-digest

- استفاده از خوشه‌بندی برای تخمین کارآمد آمار مبتنی بر رتبه در جریان‌های نامحدود

نمونه‌برداری تصادفی

نمونه‌برداری تصادفی

- انتخاب بدون جایگزینی زیرمجموعه‌ای تصادفی از داده‌ها
- مورد استفاده در بسیاری از الگوریتم‌های علم رایانش

در مسائل رتبه

- امکان استفاده از چنین روشی
- در تهیهٔ گزارش چندک‌های محاسبه شده روی نمونه‌ها
- به عنوان تقریبی از چندک‌های کل جریان داده

نمونه‌برداری تصادفی

مزیت متمایز چنین مجموعه نمونه‌هایی

- کوچکتر بودن آنها
- امکان استفاده از الگوریتم های قطعی کلاسیک در پاسخ به چنین پرس‌وجوهای چندک رتبه روی مجموعه نمونه

چالش

- با توجه به چولگی داده‌ها
- نیاز به داشتن تضمین‌های قبلی در مورد خطای چنین تقریبی،
- روشی خاص در نمونه‌برداری تصادفی
  - امکان وابستگی روش به داده‌ها

نمونه‌برداری تصادفی

مشکل دیگر نمونه‌برداری ساده

- بسیاری از طرح‌های نمونه‌برداری نیاز به اطلاع قبلی از اندازه مجموعه داده
- ناممکن یا مشکل‌آفرین در جریان‌های پیوسته معمول در برنامه‌های داده بزرگ

از راه‌حل‌های ممکن

- روش نمونه‌برداری مخزنی
  - پیشنهاد جفری ویتر در سال ۱۳۶۴
- امکان تولید نمونه بدون اطلاع از تعداد
- استفاده از آن در مسئله چندک
  - نیازمند به حافظه‌ای قابل‌توجه

نمونه‌برداری مخزنی

الگوریتم ۱- نمونه‌برداری مخزنی

ورودی: جریان داده D و میزان چندک q

ورودی: بافر خالی B با اندازه k

خروجی: بافر پر شده B

$B \leftarrow D[۱:k]$

برای i از $k+۱$ تا انتها

$r = rnd(۱, i)$

اگر $r \leq k$

B[r]=D[i]

مرتب‌سازی B

$j = \lceil q \times k \rceil$

برگرداندن B[j]

نمونه‌برداری مخزنی

[3, 1, 7, 2, 9, 4, 8, 5, 6]

K=4

B=[3, 1, 7, 2]

نمونه‌برداری مخزنی

[3, 1, 7, 2, 9, 4, 8, 5, 6]

K=4

B=[3, 1, 7, 2]

i=5

- مقدار ۹
- r=3
  - $r \leq k$
- B=[3, 1, 9, 2]

نمونه‌برداری مخزنی

[3, 1, 7, 2, 9, 4, 8, 5, 6]

K=4

B=[3, 1, 9, 2]

i=6

- مقدار 4
- r=5
  - $r > k$
- B=[3, 1, 9, 2]

نمونه‌برداری مخزنی

[3, 1, 7, 2, 9, 4, 8, 5, 6]

K=4

B=[3, 1, 9, 2]

i=7

- مقدار 8
- r=2
- $r \leq k$
- B=[3, 8, 9, 2]

نمونه‌برداری مخزنی

[3, 1, 7, 2, 9, 4, 8, 5, 6]

K=4

B=[3, 8, 9, 2]

i=8

- مقدار 5
- r=6
  - $r > k$
- B=[3, 8, 9, 2]

نمونه‌برداری مخزنی

[3, 1, 7, 2, 9, 4, 8, 5, 6]

K=4

B=[3, 8, 9, 2]

i=9

- مقدار 6
- r=1
- $r \leq k$
- B=[6, 8, 9, 2]

مرتب‌سازی

- B=[2, 6, 8, 9]
- میانه برابر ۷ در اصلی برابر ۵

نیاز به حافظهٔ $O\left(\frac{1}{\epsilon^2}\right)$

نمونه‌برداری تصادفی مرل MRL

الگوریتم نمونه‌برداری تصادفی

- MRL
- پیشنهاد گورمیت سینگ مانکو، سریدهار راجاگوپالان و بروس لیندسی
- در سال ۱۳۷۸
- جهت حل مسئله نمونه‌برداری صحیح و تخمین چندک

شامل

- روش نمونه‌برداری غیریکنواخت
- الگوریتم یافتن چندک قطعی

نمونه‌برداری تصادفی مرل

پشتیبانی از پردازش جریان داده پیوسته با نیازهای فضای کم

- پیشنهاد اصلاح غیریکنواخت از نمونه‌برداری مخزنی
- گنجاندن مقادیری که زودتر در دنباله ظاهر می‌شوند با احتمال بیشتر نسبت به سایرین
  - بهبود کارایی فضایی
  - دقیق‌تر از نمونه‌برداری مخزنی اصلی

نمونه‌برداری تصادفی مرل

معایب اصلی

- تعیین پارامترهای پیکربندی آن با حل مسئله بهینه‌سازی
  - پیچیدگی آن و استفاده از رویه‌های پیچیده

معرفی نسخه‌ای ساده از الگوریتم مرل

- پیشنهادی گه لو، لو وانگ، که یی، و گراهام کورمودی
- در سال ۱۳۹۲
- در مقالات اصلی با عنوان Random

الگوریتم ت

پردازش داده‌های جریان داده در قطعات با اندازه متغیر

نمونه‌برداری از آنها

تولید نهایی نمونه‌برداری غیریکنواخت

الگوریتم ت

استفاده از داده‌ساختار یا آرایه‌ای شخصی شده برای ذخیره نمونه‌های مقادیر

حاصل از b واحد دادهٔ $B_1, B_2, \ldots B_b$

- معروف به بافر

هر یک امکان ذخیره تا حداکثر k مقدار

مرتبط شدن با سطح L یا سطح پر شدن

الگوریتم ت

پارامتر سطح L

- نمایانگر احتمال انتخاب مقادیر

وابسته به

- تعداد مقادیر n تاکنون پردازش شده
- حداکثر ارتفاع مجاز h درخت نمایش دهندهٔ دنباله عملیات انجام شدهٔ الگوریتم

$$L = L(n,h) = \max\left(0, \left\lceil \log \frac{n}{k2^{h-1}} \right\rceil\right)$$

$$L(0,h) = 0$$

الگوریتم ت

برای پر کردن بافر خالی B_i^L
- $i \in 0 \ldots b$
- در سطح L

انتخاب k مقدار تصادفی از $k \cdot 2^L$ مقدار ورودی متوالی،

هر مقدار به ازای هر بلوک از 2^L بلوک

ذخیره مقادیر در B_i^L

در پایان رویه
- بافر کمتر از k مقدار
- به دلیل نبود مقدار به اندازهٔ کافی در دنباله ورودی
- در صورت وجود حداقل یک مقدار در بافر
  - برچسب‌گذاری به عنوان پر

الگوریتم ت

احتمال انتخاب مقداری خاص از جریان داده ورودی و ذخیره در بافر

- وابسته به سطح L
  - اندازه قطعه 2^L کنترل کنندهٔ استخراج مقادیر از آن

پیاده‌سازی عملی نمونه‌برداری غیریکنواخت مورد استفادهٔ الگوریتم

الگوریتم ۱- پر کردن بافرهای خالی

ورودی: جریان داده D

ورودی: بافر خالی B^L با اندازه k در سطح L

خروجی: بافر پر شده B^L و برچسب آن

برای i از ۰ تا ۱ - k انجام بده

$S \leftarrow next(۲^L, D)$ //خواندن ۲^L مقدار بعدی از D

اگر $S = \emptyset$ آنگاه

break

$x \leftarrow$ نمونه$(s \in S)$ //انتخاب تصادفی مقدار از S

$B^L \leftarrow B^L \cup \{x\}$

برچسب ← خالی

اگر ۰ < (B^L)تعداد آنگاه

برچسب ← پر

برگرداندن B^L، برچسب

الگوریتم ت

امکان ادغام دو بافر از یک سطح L جهت بازیابی فضای بافر

- منجر به بافری جدید با اندازه مشابه در سطح L + 1

جهت ادغام دو بافر

- مرتب‌سازی دنباله مقادیر هر دو بافر
- انتخاب تصادفی نیمی از مقادیر
  - به عنوان مثال، با انتخاب همه مقادیر در موقعیت‌های زوج یا فرد

علامت‌گذاری بافرهای ادغام شده به عنوان خالی

- علامت‌گذاری بافر خروجی به عنوان پر

الگوریتم ۲- ادغام دو بافر غیر خالی

ورودی: بافرهای غیر خالی B_i^L ، B_j^L با اندازه k در سطح L

خروجی: بافر پر شده B^{L+1} در سطح L و برچسب آن

$S \leftarrow sort(B_i^L \cup B_j^L)$

$free(B_i^L)$

$free(B_j^L)$

$B^{L+1} \leftarrow$ نمونه(S, k)// انتخاب تصادفی k مقدار از بافرهای پیوسته

برگرداندن B^{L+1}، پر

الگوریتم ت

عملیات ادغام
- مرتب‌سازی بافرها نیاز به O(klog k)
- آماده کردن جمعیت بعدی بافر در زمان O(k)

فرآیند ساخت داده‌ساختار
- شامل سری مراحل جمعیت بافر و عملیات ادغام

آغاز
- برچسب‌گذاری هر بافر به عنوان خالی

پردازش جریان ورودی با فعال‌سازی سطح L
- در ابتدا برابر با صفر به دلیل عدم وجود مقادیر پردازش شده

در صورت خالی بودن بافر خالی B
- پر کردن با استفاده از الگوریتم ۱ با خواندن $k \cdot 2^L$ مقدار از جریان

پر شدن همه بافرها
- یافتن حداقل دو بافر با پایین‌ترین سطح
- ادغام دو بافر

مثال- ساخت بافرهای نمونه

مجموعه داده از ۲۵ عدد صحیح

$\{0,0,3,4,1,6,0,5,2,0,3,3,2,3,0,2,5,0,3,1,0,3,1,6,1\}$

استفاده از ارتفاع $h = 3$

بافر $b = 4$

- B_1, B_2, B_3, B_4
  - هر یک k=4 مقدار

محاسبهٔ سطح فعال با استفاده از معادله

$$L = L(n) = max\,(0, \lceil log\,(n) - 4 \rceil)$$

مثال- ساخت بافرهای نمونه

در ابتدا،

تعداد مقادیر پردازش شده $n = 0$

پس: شروع از $L = 0$

- خواندن اولین $N_1 = 4$ مقدار از جریان ورودی $\{0,0,3,4\}$
  - پر کردن یکی از بافرهای خالی، مثلاً B_1
  - ظرفیت بافر برابر ۴ در نتیجه عدم نیاز به کنار گذاشتن مقادیر تصادفی و ذخیره همه آنها

B_1^0				B_2				B_3				B_4			
0	0	3	4												

مثال- ساخت بافرهای نمونه

دوم،

- تعریف دوبارهٔ سطح فعال
- تعداد مقادیر پردازش شده تاکنون $n = N_1 = 4$
- $L = L(4) = max\,(0,2 - 4) = 0$
  - سطح فعال برابر سطح صفر
- خواندن $N_2 = 4$ مقدار بعدی $\{1,6,0,5\}$
- و پر کردن بافر B_2 به همان ترتیب

B_1^0				B_2^0				B_3				B_4			
0	0	3	4	1	6	0	5								

مثال- ساخت بافرهای نمونه

پردازش $n = N_1 + N_2 = 8$ مقدار تاکنون

سطح فعلی برابر

- $L = L(8) = \max(0,3 - 4) = 0$
- خواندن $N_3 = 4$ مقدار بعدی $\{2,0,3,3\}$ نمایه‌سازی آنها به بافر B_3

B_1^0				B_2^0				B_3^0				B_4			
0	0	3	4	1	6	0	5	2	0	3	3				

مثال- ساخت بافرهای نمونه

پردازش $n = N_1 + N_2 + N_3 = 12$ مقدار تاکنون

سطح فعال هنوز برابر صفر

خواندن $N_4 = 4$ مقدار بعدی $\{2,3,0,2\}$ و پر کردن آخرین بافر خالی B_2

B_1^0				B_2^0				B_3^0				B_4^0			
0	0	3	4	1	6	0	5	2	0	3	3	2	3	0	2

مثال- ساخت بافرهای نمونه

پر شدن همه بافرها در این مرحله

نیاز به عملیات ادغام

پایین‌ترین لایهٔ دارای حداقل دو بافر: سطح ۰

انتخاب تصادفی دو بافر از آن سطح

- مثلا، B_2^0 و B_3^0
- ادغام و سپس مرتب‌سازی

$$\{1,6,0,5\} \cup \{2,0,3,3\} = \{1,6,0,5,2,0,3,3\} \rightarrow \{0,0,1,2,3,3,5,6\}$$

مثال- ساخت بافرهای نمونه

گام بعدی

- آزاد کردن بافرهای B_2^0 و B_3^0
- پر کردن بافر B_3 در سطح ۱ با ۵۰ درصد از مقادیر قبلی آنها
- جهت سادگی انتخاب مقادیر فرد

B_1^0				B_2				B_3^1				B_4^0			
0	0	3	4					0	1	3	5	2	3	0	2

مثال- ساخت بافرهای نمونه

تاکنون پردازش $n = N_1 + N_2 + N_3 + N_4 = 16$ مقدار

- لایه فعال باقی ماندن لایه صفر و پر کردن B_2 با $N_5 = 4$ مقدار بعدی از جریان داده $\{5,0,3,1\}$

B_1^0				B_2^0				B_3^1				B_4^0			
0	0	3	4	5	0	3	1	0	1	3	5	2	3	0	2

مثال- ساخت بافرهای نمونه

عدم وجود بافر خالی

- نیاز به ادغامی دیگر

سطح ۰ شامل سه بافر پر

انتخاب تصادفی دو مورد از آنها

- مثلا، B_1^0 و B_4^0،

ادغام مقادیر آنها و در ادامه مرتب‌سازی

$$\{0,0,3,4\} \cup \{2,3,0,2\} = \{0,0,3,4,2,3,0,2\} \rightarrow \{0,0,0,2,2,3,3,4\}$$

مثال- ساخت بافرهای نمونه

برچسب‌زنی بافرهای B_1^0 و B_4^0 به عنوان خالی

پر کردن بافر B_4 در سطح ۱ با ۵۰ درصد از مقادیر آنها با گرفتن مقادیر در موقعیت‌های زوج

B_1				B_2^0				B_3^1				B_4^1			
				5	0	3	1	0	1	3	5	0	2	3	4

مثال- ساخت بافرهای نمونه

گام بعد

تاکنون پردازش $n = N_1 + N_2 + N_3 + N_4 + N_5 = 20$

سطح فعال برابر $L = L(20) = 4.32 - 4 = 1$

خواندن $N_6 = 4 \cdot 2^1 = 8$ مقدار بعدی از جریان داده

- باقی نماندن مقادیر کافی در جریان داده
- خواندن $\{0,3,1,6,1\}$
- پرکردن B_1 با نمونه‌برداری یک مقدار از هر گروه دو مقداری

B_1^1				B_2^0				B_3^1				B_4^1			
3	1	1		5	0	3	1	0	1	3	5	0	2	3	4

پرسش چندک معکوس

امکان پاسخ به پرسش چندک معکوس با استفاده از بافر نمونه

تخمین رتبه مقدار داده شده **x** به صورت مجموع وزنی لایه از شمارش مقادیر کوچکتر از **x** برای هر بافر غیر خالی

$$rank(x) = \sum_{i=1}^{b} 2^{L(B_i)} \cdot \left|\left\{e < x : e \in B_i^L(B_i)\right\}\right|$$

مثال- پرسش چندک معکوس با نمونه‌برداری تصادفی

جریان داده از مثال قبلی

پرس‌وجوی چندک معکوس با تخمین رتبه مقدار ۴

داده‌ساختار به بافر نمونه

B_1^1				B_2^0				B_3^1				B_4^1			
3	1	1		5	0	3	1	0	1	3	5	0	2	3	4

مثال- پرسش چندک معکوس با نمونه‌برداری تصادفی

B_1^1				B_2^0				B_3^1				B_4^1			
3	1	1		5	0	3	1	0	1	3	5	0	2	3	4

با استفاده از

$$rank(x) = \sum_{i=1}^{b} 2^{L(B_i)} \cdot \left|\{e < x : e \in B_i^L(B_i)\}\right|$$

محاسبهٔ

$$rank(4) = 2^1 \cdot 3 + 2^0 \cdot 3 + 2^1 \cdot 3 + 2^1 \cdot 3 = 21$$

تخمین رتبهٔ مقدار ۴ رتبه $\text{rank}(4) = 21$

پرس‌وجوی چندک و یافتن Q-چندک

جهت حل پرسش چندکی و یافتن q-چندک

- از داده‌ساختار بافر نمونه

در پی مقداری با بدست آوردن رتبه تخمینی آن حاصل از معادله

- نزدیک‌ترین به $q \cdot n$

امکان محاسبه با تعدادی پرس‌وجوی چندک معکوس برای هر یک از مقادیر در داده‌ساختار

روش دیگر

- استفاده از جستجوی دودویی برای تسریع فرآیند
- توقف به محض یافتن مقدار به اندازه کافی نزدیک

مثال- پرس‌وجوی چندک با نمونه‌برداری تصادفی

جریان داده مثال قبلی

محاسبهٔ چندک ۰٫۶۵

تعداد کل اعداد در داده‌ساختار $n = 25$

- بنابراین مقدار مرزی حدود $q \cdot n = 0.65 \cdot 25 = 16.25$

B_1^1				B_2^0				B_3^1				B_4^1			
3	1	1		5	0	3	1	0	1	3	5	0	2	3	4

مثال- پرس‌وجوی چندک با نمونه‌برداری تصادفی

وجود مقادیر $\{0,1,2,3,4,5\}$ در بافر نمونه

آغاز با مقدار ۰

- تخمین رتبه آن
  - برابر با صفر $rank(0) = 0$

سپس، بررسی مقدار ۱

- تخمین رتبه آن $rank(1) = 5$

رتبه مقدار ۲ برابر $rank(2) = 12$

رتبه مقدار ۳ برابر $rank(3) = 14$

از مثال قبل رتبهٔ ۴ برابر $rank(4) = 21$

رتبه مقدار ۵ برابر $rank(5) = 23$

مثال- پرس‌وجوی چندک با نمونه‌برداری تصادفی

نزدیک‌ترین مقدار به مقدار مرزی ۱۶,۲۵ مقدار ۳ با rank(3) = 14

گزارش مقدار ۳ به عنوان تقریبی از چندک ۰,۶۵

امکان تسریع فرایند با استفاده از جستجوی دودویی روی مجموعه مرتب شده اعداد،

- با در نظر گرفتن اینکه رتبه تابعی یکنواخت از آرگومان خود

ویژگی‌ها

برای محاسبهٔ ϵ- تقریبی q - چندک،

نیاز به مقدار ثابتی از حافظه

- متناسب با $b \cdot k$
- فقط وابسته به ϵ

با خطای تقریب داده شده، امکان داشتن درخت محاسباتی با ارتفاع $h = \log\frac{1}{\epsilon}$

تعداد بهینه بافرها

$$b = \log\frac{1}{\epsilon} + 1,$$

اندازه هر بافر

$$k = \frac{1}{\epsilon}\sqrt{\log\frac{1}{\epsilon}}$$

ویژگی‌ها

کراندار بودن تصادفی بودن الگوریتم ت تقریب‌های چندک با احتمال خطای ثابت با $\frac{1}{2}\epsilon$

ناشی از نمونه‌برداری تصادفی و مراحل ادغام تصادفی

الگوریتم ت

تعداد کل عملیات ادغام در کل جریان داده $O\left(\frac{n}{k}\right)$

- هر به‌روزرسانی حدود $O(1)$

مرتب‌سازی هر به‌روزرسانی O(log k)

زمان سرشکنی O(log k)

Q-DIGEST

خلاصه چندک

- یا q – digest

الگوریتم خلاصه جریان مبتنی بر درخت

نیشیت شریواستاوا، چیرانجیب بوراگوهین، دیویاکانت آگروال و سابهاش سوری

در سال ۱۳۸۳

پیشنهاد جهت استفاده در زمینه نظارت بر داده‌های توزیع شده از حسگرها

تلخیص چ Q-DIGEST

تبدیل یا حل مسئله محاسبه چندک

- به عنوان مسئله هیستوگرام
- تلخیص داده‌ها به تعدادی سطل

نگه‌داری مجموعه‌ای از چنین سطل‌هائی در داده‌ساختاری درختی

ادغام سطل‌های کوچک و تقسیم سطل‌های بزرگ

الگوریتمی «قطعی» با «اتلاف»

تلخیص چ

اجرای الگوریتم با مقادیر صحیح در بازه‌ای از پیش معین

تقسیم‌بندی دودویی بازهٔ صحیح [0, N - 1]

- به مثابهٔ درخت دودویی کامل
- مقدار ریشه آن شامل کل بازه [0, N - 1]
- فرزندان چپ و راست آن دارای بازهٔ $\left[0, \left\lfloor \frac{N-1}{2} \right\rfloor\right]$ و $\left[\left\lfloor \frac{N-1}{2} \right\rfloor + 1, N - 1\right]$

گره‌های برگ نمایش دهندهٔ مقادیر صحیح منفرد

عمق درخت log N

تلخیص چ

هر گره v از درخت دارای

- بازهٔ متناظر $[v_{\text{کم}}, v_{\text{بیش}}]$
- شمارندهٔ متناظر شمارندهv
  - نمایش تعداد اعضای سطل (شامل تکراری‌ها)

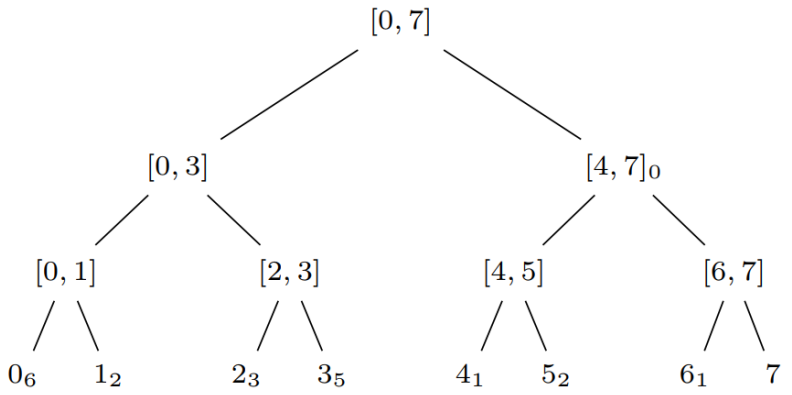
مثال- تقسیم‌بندی دودویی در تلخیص چ

مجموعه داده‌ای از $n = 20$ عدد صحیح از محدوده $[0,7]$:
$\{0,0,3,4,1,6,0,5,2,0,3,3,2,3,0,2,5,0,3,1\}$

ایجاد درخت دودوئی با تقسیم‌بندی دودویی بازه

سطل‌بندی داده‌های ورودی

- هر برگ شامل مقدار و بسامد



تلخیص چ

گره‌های داخلی داده شامل بسامدهای مقادیر ذخیره شده متناظر

در بدترین حالت ذخیره‌سازی

- ذخیرهٔ داده‌ها به صورت O(n) یا O(N)
  - هر کدام کوچکتر

احتمال بالای تنکی و نامتوازنی چنین درخت‌های دودویی

- ⟸ ناکارآمدی ذخیره آن به صورت خام بدون فشرده‌سازی

تلخیص چ

پیشنهاد روش فشرده‌سازی و ذخیره فشرده چنین درخت تقسیم دودویی

داده‌ساختار تلخیص چ

- کدگذاری اطلاعات مربوط به توزیع مقادیر
- نمایش نسخه‌ای از درخت دودویی شامل سطل‌های ν با ویژگی زیر

$$\begin{cases} \nu_{count} \leq \lfloor n\sigma \rfloor, & \text{به جز سطل‌های برگ} \\ \nu_{count} + \nu^{p}_{count} + \nu^{s}_{count} > \left\lfloor \frac{n}{\sigma} \right\rfloor, & \text{به جز ریشه} \end{cases}$$

- ν^{p} والد و ν^{s} همزاد ν
- n تعداد کل عناصر
- $\sigma \in [1,n]$ پارامتر طراحی و مسئول سطح فشرده‌سازی

تلخیص چ

سطل‌های ریشه و برگ

- دارای استثنا

گنجاندن ریشه در داده‌ساختار

گنجاندن سطل‌های برگ با شمارنده‌های بزرگتر از مقدار مرزی $\left\lfloor \frac{n}{\sigma} \right\rfloor$ (مقادیر پربسامد)

ویژگی تلخیص سبک سنگینی بین گنجاندن چند سطل سطح بالا و گسترده، و بسیاری از سطل‌های کوچک حاوی اطلاعاتی در مورد چند مقدار کم‌بسامد

تلخیص چ

ویژگی نخست

- $\nu_{count} \leq \lfloor n\sigma \rfloor$

- حذف کننده سطل‌ها مگر گره‌های برگ با شمارنده‌های مقادیر با بسامد بالا
- به دلیل نیاز به دقت بیشتر و ارزش ذخیرهٔ چنین سطل‌هایی ارزش ذخیره مقادیر فرزند و داشتن شمارنده‌های دقیق‌تر

ویژگی دوم

- $\nu_{count} + \nu^{p}_{count} + \nu^{s}_{count} > \left\lfloor \frac{n}{\sigma} \right\rfloor$
- در صورت همزادی دو سطل دارای شمارندهای کوچک
- در پی اجتناب از داشتن دو شمارنده جداگانه برای هر یک از آنها
- توصیه به ادغام آنها در والد خود
  - منجر به نائل شدن به درجه فشرده‌سازی مورد نیاز

تلخیص چ

سخن کوتاه

- ایجاد داده ساختار نیازمند ادغام سلسله‌مراتبی و کاهش سطل‌ها
- گذر از تمام سطل‌ها از پایین به بالا
- بررسی نقض ویژگی در هر یک

در عمل بررسی محدودیت دوم

- به دلیل گذر پائین به بالا

تلخیص چ

به جز سطل ریشه،

هر سطل v ناقض ویژگی تلخیص

- فشرده‌سازی زیردرخت آن
- از طریق فشرده‌سازی شمارنده‌های آن
- ادغام والد v^p و همزاد v^s
- ارتقاء آنها به سطل والد

$$v^p_{count} = v^p_{count} + v_{count} + v^s_{count}$$

حذف سطل v و همزاد آن v^s از تلخیص

تلخیص چ- فشرده‌سازی

الگوریتم ۴- فشرده‌سازی تلخیص چ

ورودی: داده‌ساختار q-digest بازهٔ $[۰, N - ۱]$

ورودی: ضریب فشرده‌سازی σ

خروجی: داده‌ساختار q-digest فشرده شده

سطح $\leftarrow log\ N - ۱$

تازمانی که (سطح > ۰) انجام بده

برای [سطح]$Q - DIGEST$ $v \in$ انجام بده

اگر $\left|\frac{n}{\sigma}\right| \leq v_{count} + v^{p}_{count} + v^{s}_{count}$ آنگاه

$v^{p}_{count} \leftarrow v^{p}_{count} + v_{count} + v^{s}_{count}$

$Q - DIGEST \leftarrow Q - DIGEST \setminus \{v, v^{s}\}$

سطح $\leftarrow$ سطح $- ۱$

برگرداندن Q-DIGEST

تلخیص چ-فشرده‌سازی

زمان فشرده‌سازی $O(m \cdot log\ N)$

- $m = |Q - DIGEST|$ تعداد سطل‌ها در داده‌ساختار

هزینه به‌روزرسانی به ازای هر مقدار حدود O(log N)

در عمل، مصرف زمان بیشتر در به‌روزرسانی

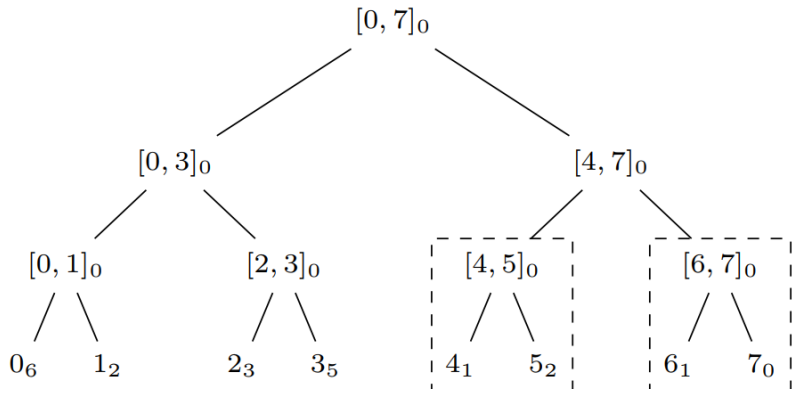
- به دلیل درج هر مقدار جدید در گره برگ
- سپس یافتن موقعیت اصلی در طول عملیات فشرده‌سازی

یافتن موقعیت مناسب خود در تلخیص چ با حرکت تک گامی مقدار در هر زمان

مثال - فشرده‌سازی درخت با تلخیص چ

مجموعه داده با $n = 20$ مقدار از مثال قبل

بسامدها برای سطل‌های مشاهده نشده به طور پیش‌فرض برابر صفر



مثال – فشرده‌سازی درخت با تلخیص چ

فشرده‌سازی با $\sigma = 5$

مقدار مرزی برابر

- $\left\lfloor \frac{n}{\sigma} \right\rfloor = \left\lfloor \frac{20}{5} \right\rfloor = 4$

حرکت از پایین به بالا،

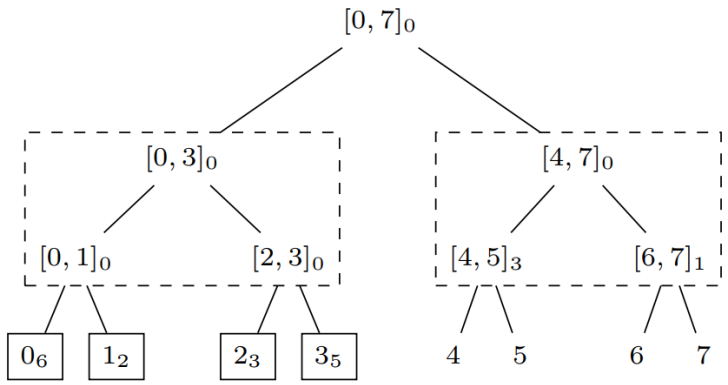
ابتدا سطح چهارم

- برآورده شدن شرط دوم تلخیص در سطل‌های ۰ تا ۳
- نقض ویژگی مذکور در فرزندان سطل‌های [۴, ۵] و [۶, ۷]
  - ادغام در والدین خود
  - حذف از داده‌ساختار

مثال - فشرده‌سازی درخت با تلخیص چ

داده ساختار پس از اعمال مراحل

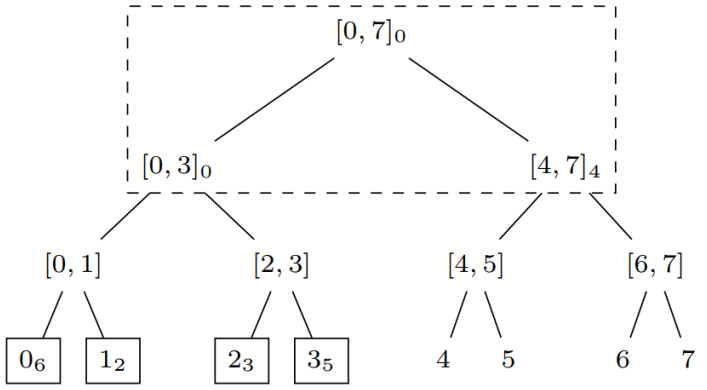
- قرارگیری سطل‌های موجود در جعبه‌ها در داده‌ساختار فشرده



مثال - فشرده‌سازی درخت با تلخیص چ

سطح سوم،

- نقض ویژگی‌های تلخیص در همه سطل‌ها
- فشرده کردن آنها در والدین خود و حذف از داده‌ساختار



مثال - فشرده‌سازی درخت با تلخیص چ

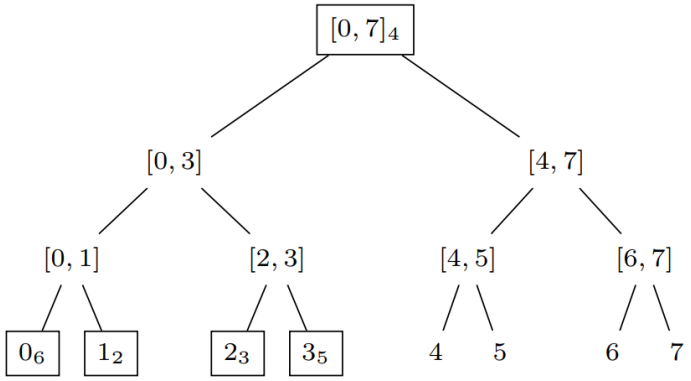
سطح دوم،

- نقض ویژگی تلخیص در دو فرزند سطل ریشه
- عدم تجاوز تعداد کل آنها از مقدار مرزی
- در نتیجه، نیاز به ادغام در والد

عدم نیاز به بررسی ریشه

- گنجانده شدن ریشه در داده ساختار اگر شمارش‌های غیرصفر مرتبط داشته باشد.

نسخه نهایی داده‌ساختار فشرده



مثال - فشرده‌سازی درخت با تلخیص چ

داده‌ساختار نیازمند ذخیره فقط پنج سطل با شمارش‌های غیرصفر

تلخیص چ-فشرده‌سازی

حرکت از پایین به بالا

یکبار تصمیم‌گیری دربارهٔ ادغام سطل‌ها در طول رویه

عدم لزوم رعایت ویژگی تلخیص در همه سطل‌های تلخیص چ فشرده پس از فشرده‌سازی

تغییرات (به عنوان مثال، ادغام در والد) در برخی از سطل‌ها در سطوح بالای درخت

- امکان نقض ویژگی با سطل‌های قبلاً گنجانده شده
- در عمل، عدم کاهش دقت با این رفتار
- در بدترین حالت، تولید داده‌ساختاری با حدی کمتر از مطلوب بودن
  - مصرف حافظه بیشتر نسبت به نظر

الگوریتم تلخیص چ

الگوریتم ۵- الگوریتم تلخیص چ

ورودی: مجموعه داده D یا مقادیری از بازهٔ $[۰, N - ۱]$

ورودی: ضریب فشرده‌سازی σ

خروجی: داده‌ساختار q-digest فشرده

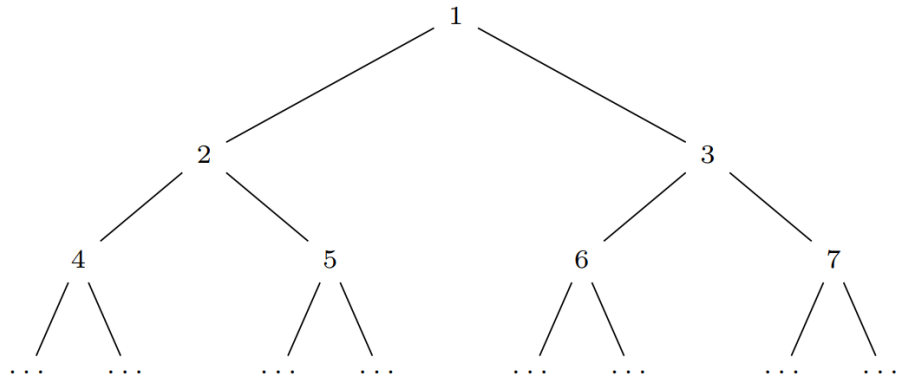
$Q - DIGEST \leftarrow BinaryPartitionTree(D, [۰, N - ۱])$

برگرداندن $Compress(Q - DIGEST, N, \sigma)$

الگوریتم تلخیص چ

بهینه‌سازی نمایش داده‌ساختار

شماره‌گذاری سطل‌ها در درخت دودویی مرتبط با روشی از راست به چپ، از بالا به پایین



الگوریتم تلخیص چ

شماره‌گذاری همه سطل‌ها

- سادگی بازیابی بازهٔ متناظر $\left[\nu_{\text{کم}}, \nu_{\text{بیش}}\right]$ با وجود اطلاع صرف از نمایه و اندیس **i** متناظر

الگوریتم ۶- بازیابی محدوده سطل $\left[\nu_{\text{کم}}, \nu_{\text{بیش}}\right]$

ورودی: نمایهٔ سطل i

خروجی: بازهٔ سطل

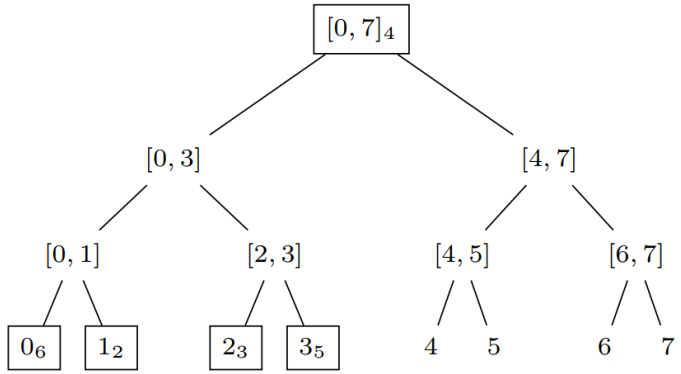
سطح $\leftarrow \lfloor log(i) \rfloor$

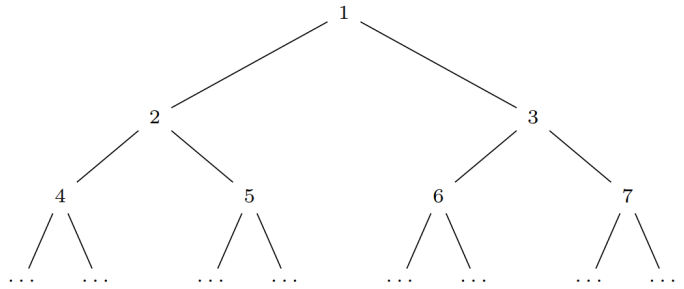
$n \leftarrow 2^{\text{سطح}}$ // تعداد سطل‌ها در سطح

$m \leftarrow i - n$ // موقعیت سطل در سطح

برگرداندن $\left\lceil \frac{(N+1)m}{n} \right\rceil - 1, \left\lceil \frac{(N+1)(m+1)}{n} \right\rceil$

نمایش خطی از داده‌ساختار





امکان ایجاد نمایش خطی از داده‌ساختار

- آرایه‌ای از سطل‌ها،
- هر سطل یک دو-تایی از شماره و شمارنده‌های مرتبط است.

مثال،

- نمایش خطی داده‌ساختار فشردهٔ مثال قبل
- $\langle(1,4),(8,6),(9,2),(10,3),(11,5)\rangle$

ادغام دو تلخیص چ

امکان ادغام دو تلخیص با ضریب فشرده‌سازی σ و محدوده‌های مقدار یکسان

فراهم‌سازی امکان پردازش توزیعی جریان‌های داده بزرگ

محاسبهٔ اتحاد مجموعه‌های سطل‌های ذخیره شده آنها

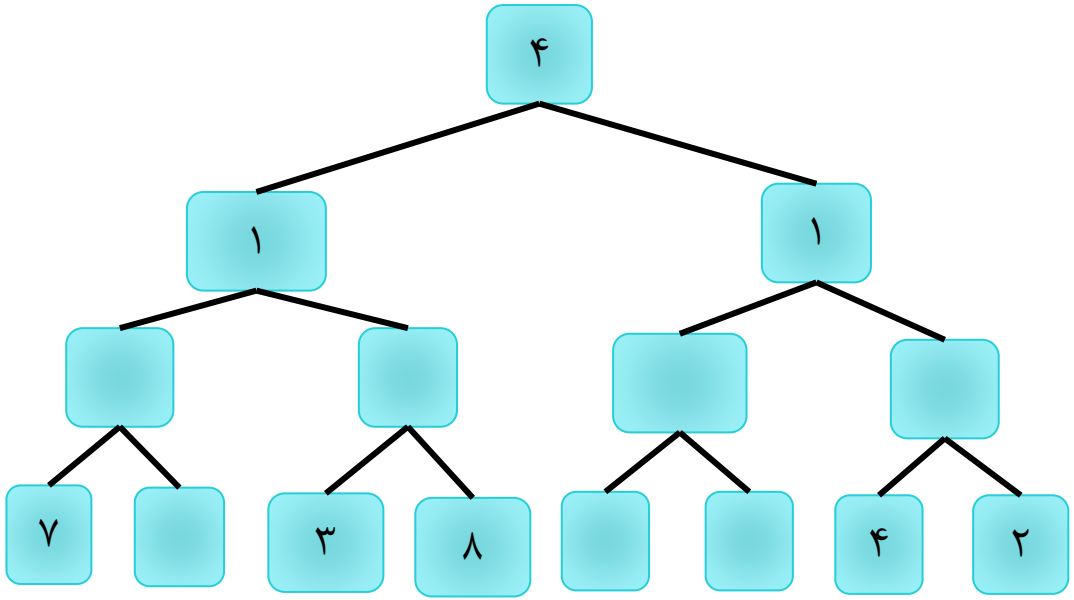
شمارش‌های سطل‌های با محدوده یکسان

جمع تعداد کل مقادیر

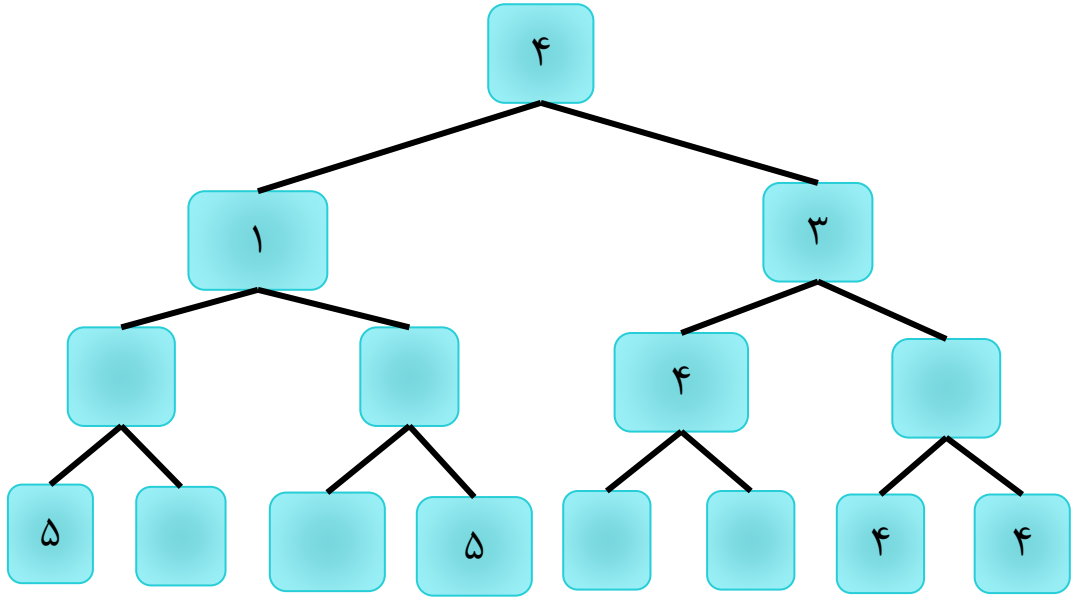
اجرای الگوریتم فشرده‌سازی

مثال-ادغام دو تلخیص چ

$$n_0 = 30, \sigma_0 = 6, \frac{n_0}{\sigma_0} = 5$$

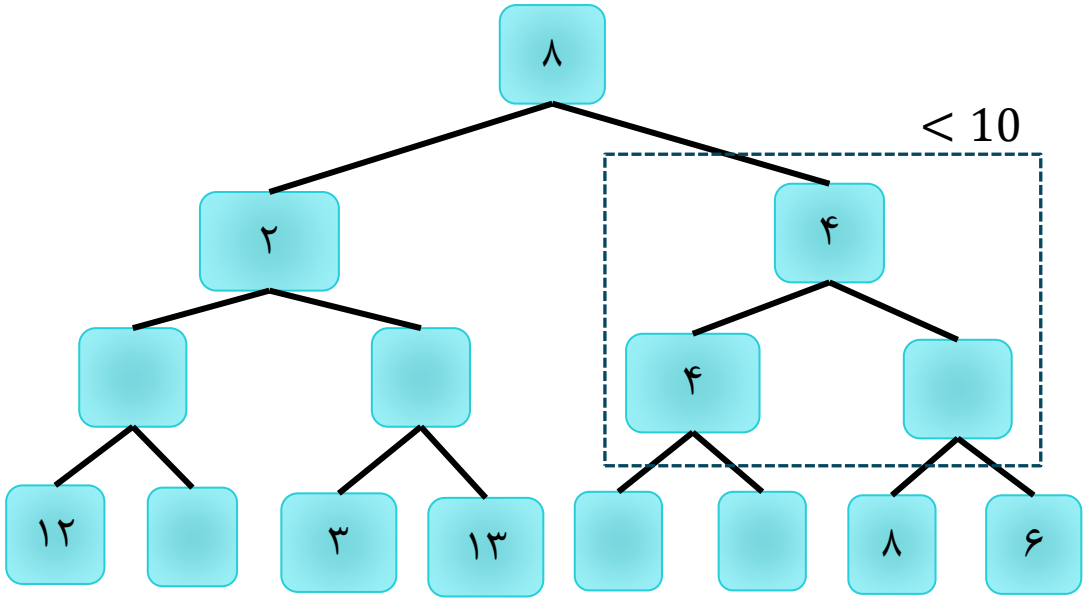


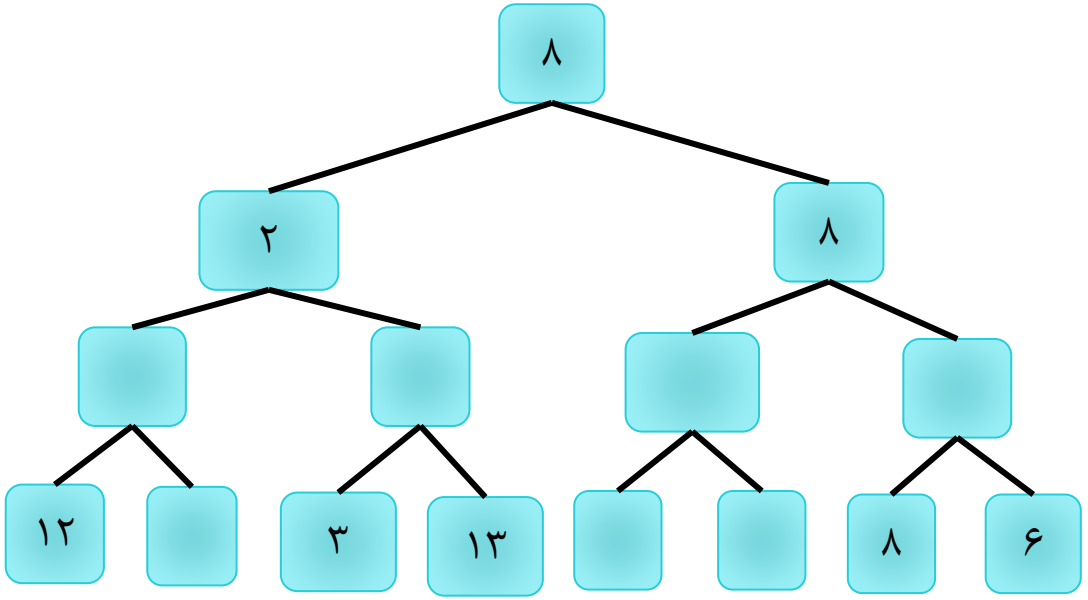
$$n_1 = 30, \sigma_1 = 6, \frac{n_1}{\sigma_1} = 5$$



مثال-ادغام دو تلخیص چ

$$n = n_0 + n_1 = 60, \sigma = 6, \frac{n}{\sigma} = 10$$





تلخیص چ-

استفاده از الگوریتم تلخیص چ در پاسخ به پرس‌وجوی چندک و یافتن q - چندک

دنباله مرتب شده S با مرتب‌سازی سطل‌ها به ترتیب صعودی مقادیر بیشνآنها

هنگام تساوی: مقادیر کوچکتر اول

پویش دنباله S از ابتدا

افزودن شمارنده‌های سطل‌های دیده شده

شرط پایان: بزرگتر شدن مقدار سطل ν^* مجموع از qn،

- تخمین رتبه سطل

بیشν^* به عنوان تخمین q - چندک

تلخیص چ- پرسش چندک

الگوریتم ۷- پاسخ به پرس‌وجوهای چندک با تلخیص چ

ورودی: داده‌ساختار q-digest

ورودی: مقدار $q \in [۰,۱]$

خروجی: q - چندک

$S \leftarrow sort(Q - DIGEST)$

$rank \leftarrow ۰$

برای $(v, count) \in S$ انجام بده

$rank \leftarrow rank + count$

اگر $rank \geq q \cdot n$ آنگاه

برگرداندن v_{max}

تلخیص چ- پرسش چندک

وجود حداقل $q \cdot n$ سطل با مقدار حداکثر کمتر از بیش$^*\nu$

- در نتیجه رتبه سطل $^*\nu$ حداقل $q \cdot n$

امکان خطا در محاسبه تقریب-ϵ چندک-q

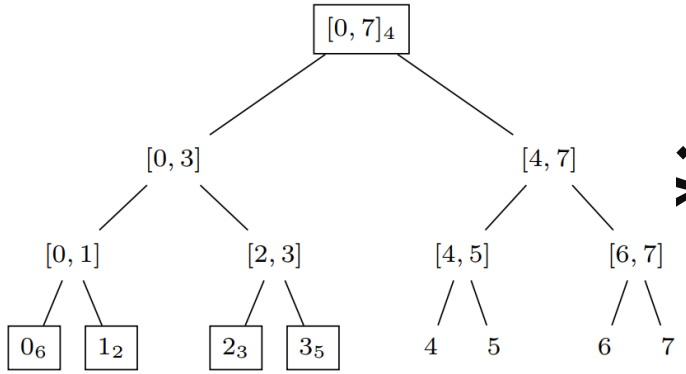
- در صورت وجود مقادیر کمتر از بیش$^*\nu$ در اجداد سطل $^*\nu$
- به دلیل شمرده نشدن در الگوریتم ۷

کران چنین خطایی با $\epsilon \cdot n$

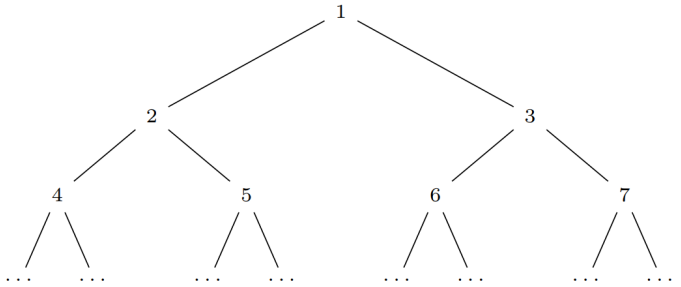
گزارش رتبه‌ای در بازه $[q \cdot n, (q + \epsilon) \cdot n]$ در الگوریتم

عدم تخمین مقدار دقیق q - چندک کمتر از حد تخمین

مثال – پرس‌وجوی چندک با تلخ



پرس‌وجوی چندک با محاسبه چندک ۰٫۶۵ از داده‌ساختار تلخیص چ مثال قبلی با نمایش خطی



$$\langle (1,4),(8,6),(9,2),(10,3),(11,5) \rangle$$

بنابراین، دنباله مرتب شده سطل‌ها

$$S = \langle (8,6),(9,2),(10,3),(11,5),(1,4) \rangle$$

مثال - پرس‌وجوی چندک با تلخیص چ

با شروع از ابتدا

جمع شمارش‌های سطل‌ها تا هنگام بزرگتر شدن مجموع از $0.65 \times n = 13$

فراتر رفتن تلخیص چ از مقدار مرزی در سطل (۱۱،۵)

- مربوط به مقدار برگ ۳

تخمین چندک ۰٫۶۵ (یا صدک شصت و پنجم) مجموعه داده مثال قبل برابر مقدار ۳

تلخیص چ- پرسش چندک معکوس

به روشی مشابه

ایجاد دنباله مرتب شده S از سطل‌ها

گذر از آنها با نگه‌داری مجموع جاری شمارش‌های سطل‌های مشاهده شده

رتبه مقدار داده شده x

- مجموع شمارش‌های سطل‌های ν با بیش $x > \nu$

تلخیص چ- پرسش چندک معکوس

الگوریتم ۸- پاسخ به پرس‌وجوهای چندک معکوس با تلخیص چ

ورودی: مقدار x

ورودی: داده‌ساختار q-digest

خروجی: رتبه عنصر

$S \leftarrow sort(Q - DIGEST)$

رتبه $\leftarrow$.

برای $(v, count) \in S$ انجام بده

اگر $x > v_{max}$ آنگاه

رتبه $\leftarrow$ رتبه $+ count$

برگرداندن رتبه

تلخیص چ- پرسش چندک بازه

الگوریتم ۹- پاسخ به پرس‌وجوهای بازه با تلخیص چ

ورودی: بازهٔ [a, b]

ورودی: داده‌ساختار q-digest

خروجی: تعداد مقادیر در محدوده

$r_a \leftarrow InverseQuantileQuery(a, Q - DIGEST)$

$r_b \leftarrow InverseQuantileQuery(b, Q - DIGEST)$

برگرداندن $r_b - r_a$

تلخیص چ- پرسش چندک معکوس

رتبهٔ حاصل در بازه $[rank(x), rank(x) + \epsilon \cdot n]$

پاسخ به پرس‌وجوی بازه

- کفایت محاسبهٔ دو پرس‌وجوی چندک معکوس برای یافتن رتبه‌ها و تفاوت بین مرزهای محدوده a و b
- حداکثر خطا برای پرس‌وجوی محدوده در تلخیص چ
  - $2\epsilon \cdot n$

ویژگی‌ها

الگوریتم با اتلاف

فشرده‌سازی اطلاعات مربوط به مقادیر با بسامد پایین

نگه‌داری اطلاعات مربوط به مقادیر با بسامد بالا

طرح تقریب خوب

- با رخ دادن تغییرات گسترده در بسامد مقادیر مختلف

الگوریتم عرضهٔ اطلاعاتی در مورد توزیع مقادیر مقادیر

- بدون در اختیار نهادی اطلاعاتی در مورد مکان رخ دادن مقادیر

ویژگی‌ها

سبک سنگینی بین دقت الگوریتم و حافظه مورد نیاز برای ذخیره داده‌ساختار

کنترل با ضریب فشرده‌سازی σ

برای محدوده داده شده [0, N]،

- انتظار ذخیرهٔ حداکثر $3 \cdot \sigma$ سطل
- خطا در محاسبه تقریب-ϵ چندک-q با کران بالای زیر

$$\epsilon \leq \frac{\log N}{\sigma}$$

ویژگی‌ها

ویژگی اصلی q-digest

- تطبیق یافتن با داده‌ها
- ایجاد سطل‌هایی با وزن تقریباً مساوی

در مقابل هیستوگرام سنتی، q-digest

- تلخیص چ اجازه همپوشانی سطل‌ها
- پاسخ به پرس‌وجوهای اجماع (به عنوان مثال، مقادیر پربسامد)

مشکلات اصلی کاربردهای عملی

- مدیریت مقادیر عدد صحیح
- نیاز به دانستن بازهٔ آنها از قبل
- خطاهای بزرگ برای چندک‌های خیلی بزرگ

تلخیص ت T-DIGEST

از روش‌های انباشت دقیق آمار مبتنی بر رتبه به صورت برخط

تد دانینگ و اوتمار ارتل در سال ۱۳۹۳

امکان تخمین چندک در جریان‌های نامحدود با تمرکز بر مقادیر خیلی بزرگ مانند چندک ۰,۹۹

باز بودن تحقیق و وجود به‌روزرسانی‌های جدید

تلخیص ت

تلخیص جریان داده ورودی D در خوشه‌هایی با اندازه متغیر $\{C_i\}_{i=1}^{m}$

منجر به حفظ مناسب دقت در محاسبه چندک در حین پردازش حجم زیادی از داده

هر خوشه C_i نمایش زیرمجموعه‌ای از مقادیر ورودی

- با اندازه‌ای جهت اطمینان حاصل
  - از بزرگ نبودن برای تخمین چندک‌ها با درون‌یابی
  - خیلی کوچک نبودن برای جلوگیری از داشتن خوشه‌های زیاد

تلخیص ت

هر خوشه C_i با مرکز جرم c_i،

- نقطه داده در مرکز خوشه،
- میانگین مقادیر ورودی
- c_i^{count} تعداد عناصر

داده‌ساختار تلخیص ت آرایه‌ای از مراکز جرم $\{(c_1,c_1^{count}),(c_2,c_2^{count}), \dots ,(c_m,c_m^{count})\}$

- مرتب شده با ترتیبی صعودی
- امکان تخمین حداکثر مقدار چندک متناظر با هر مرکز جرم c_i

$$q(c_i) = \frac{1}{n}\left(\sum_{j<i} c_j^{count} + c_i^{count}\right)$$

$$q(c_i) = \frac{1}{n}\left(\sum_{j<i} c_j^{count} + \frac{1}{۲}\cdot c_i^{count}\right)$$

- $n = \sum_{j=1}^{m} c_j^{count}$ تعداد کل مقادیر نمایه شده در داده‌ساختار

تلخیص ت

هر خوشه C_i در داده‌ساختار مسئول بازهٔ خاصی از مقادیر چندک $[q(c_{i-1}),q(c_i)]$

وابستگی طول آن به اندازه خوشه

تعیین اندازه صحیح خوشه تأثیر مستقیم بر دقت

انجام با تابع مقیاس غیرنزولی

تابع $k = k(q,\sigma)$ در نظر گرفتن فشرده‌سازی مطلوب σ و مقادیر چندک q بر اساس فاصله از انتها

- مانند $q = 0$ و $q = 1$

اهمیت انتخاب خوب تابع مقیاس

- وجود توابع متفاوت از نظر دقت
- مثال، از توابع رایج

$$k(q,\sigma) = \frac{\sigma}{2}q$$

$$k(q,\sigma) = \frac{\sigma}{2\pi}arcsin\,(2q - 1)$$

- پارامتر فشرده‌سازی $\sigma > 1$
- مقادیر بزرگتر مربوط به فشرده‌سازی کمتر

تلخیص ت

با توجه به تابع مقیاس انتخاب شده $k = k(q,\sigma)$،

مرتبط بودن هر خوشه C_i با مرکز جرم c_i در داده‌ساختار

تعریف اندازه k (یا $K(c_i)$) با طول مقیاس‌گذاری شده بازهٔ چندک برای خوشه

$$K(c_i) = k(q(c_i),\sigma) - k(q(c_{i-1}),\sigma), i = 2 \dots m$$

- $K(c_1) = k(q(c_1),\sigma)$

تلخیص ت

برای محدود کردن تعداد مقادیر در خوشه

- امکان محدودسازی اندازه k
- با توابع مقیاس غیرخطی،
  - امکان وجود خوشه‌های غیریکنواخت، با اندازه‌های خوشه بزرگتر برای چندک‌های محدوده میانی و کوچکتر در نزدیکی انتها
  - تا خوشه‌های تکی صرفا حاوی تک مقدار

طراحی الگوریتم برای ساخت داده‌ساختار کاملاً ادغام‌پذیر

- همه خوشه‌ها $\{C_i\}_{i=1}^{m}$ برآورده‌ساز «ویژگی تلخیص»

$$\begin{cases} K(c_i) \leq 1, & \text{به جز خوشه‌های تکی} \\ K(c_i) + K(c_{i+1}) > 1 \end{cases}$$

محدودساز اندازهٔ k هر خوشه

تضمین عدم امکان ادغام بیشتر هر دو خوشه مجاور

تلخیص ت

در عمل، جهت برآورده‌سازی محدودیت‌های ویژگی تلخیص

- عدم‌نیاز به محاسبه مجدد اندازهٔ k برای هر خوشه در هر تغییر
- به دلیل مرتب بودن همه مراکز جرم
  - امکان محدود کردن تعداد مقادیر در خوشه C_i با انتخاب مرزی برای مقدار چندک حداکثر تخمینی

$$q_{limit} = k^{-1}(k(q(c_i), \sigma) + 1, \sigma)$$

- مقدار آن برای تابع مقیاس قبلی

$$q_{limit} = \frac{1}{2}\left[1 + sin\left(\arcsin(2 \cdot q(c_i) - 1) + \frac{2\pi}{\sigma}\right)\right]$$

تلخیص ت

با داشتن قوانین محدود کردن تعداد مقادیر در هر خوشه

- امکان تعریف الگوریتم ادغام
- مشابه رویه خوشه‌بندی منظم

خلاصه کردن دنباله ورودی از نقاط داده وزنی $\{(x_1,x_1^{count}),(x_2,x_2^{count}), \dots ,(x_b,x_b^{count})\}$

- مرتب‌سازی آنها همراه با همه مراکز جرم از داده‌ساختار
- با گذری یکباره از دنباله حاصل X، ادغام آنها در صورت عدم نقض ویژگی تلخیص
- آغاز با چپ‌ترین مرکز جرم
- در نظر گرفتن خوشه متناظر به عنوان خوشه نامزد فعلی
- محاسبهٔ مقدار مرزی آن q_{limit}
- تخمین مقادیر چندک تقریبی آنها با پردازش متوالی همه مراکز جرم از X
- مقایسه با مقدار مرزی
- در صورت عدم تجاوز میانگین جذب مرکز جرم از مقدار مرزی، ادغام آن در خوشه نامزد
- ادامه با مرکز جرم بعدی از دنباله X
- در غیر این صورت، به معنای رسیدن به حداکثر ظرفیت خوشه نامزد
- عدم امکان افزودن مقادیر جدید
- ذخیرهٔ خوشه نامزد فعلی در داده‌ساختار
- انتشار خوشه نامزد جدید با مرکز جرم میانگین
- محاسبهٔ دوباره مقدار مرزی چندک q_{limit}
- در پایان، حاصل داده‌ساختار کاملاً ادغام شده

تلخیص ت

الگوریتم ۱۰- ادغام مقادیر در تلخیص ت

ورودی: بافر B با مقادیر $\{(x_1, x_1^{count}), (x_2, x_2^{count}), \dots (x_b, x_b^{count})\}$

ورودی: داده‌ساختار T-DIGEST

ورودی: پارامتر فشرده‌سازی $\sigma > 1$، تابع مقیاس k

خروجی: داده‌ساختار t-digest با بافر ادغام شده

$X \leftarrow sort(T - DIGEST \cup B)$

$T - DIGEST \leftarrow \emptyset$

$m \leftarrow count(X), n \leftarrow \sum_{x_i \in X} x_i^{count}$

$c \leftarrow x_1, q_c \leftarrow 0$

$q_{limit} \leftarrow k^{-1}(k(q_c, \sigma) + 1, \sigma)$

برای i از ۲ تا m انجام بده

$\hat{q} \leftarrow q_c + \frac{1}{n} c^{count} + \frac{1}{n} \cdot x_i^{count}$

اگر $\hat{q} \leq q_{limit}$ آنگاه

$c^{count} \leftarrow c^{count} + x_i^{count}$

$c \leftarrow c + x_i^{count} \cdot \frac{x_i - c}{c^{count} + x_i^{count}}$

continue

$T - DIGEST \leftarrow T - DIGEST \cup \{(c, c^{count})\}$

$q_c \leftarrow q_c + \frac{1}{n} c^{count}$

$q_{limit} \leftarrow k^{-1}(k(q_c, \sigma) + 1, \sigma)$

$c \leftarrow x_i$

$T - DIGEST \leftarrow T - DIGEST \cup \{(c, c^{count})\}$// آخرین خوشه معمولا تکی

برگرداندن T-DIGEST

تلخیص ت

با هر بار انتشار خوشه نامزد جدید

- نیاز به محاسبهٔ دوبارهٔ مقدار مرزی q_{limit}
- شامل محاسبه پرهزینه تابع مقیاس و معکوس آن

کم بودن تعداد خوشه‌ها در عمل

معرفی تکنیک‌های مختلف برای بهینه‌سازی تخمین مرزی

- مانند استفاده از تقریب‌های کارآمد برای اجزای توابع مقیاس یا تخمین تقریبی حداکثر تعداد عناصری که می‌توان در هر خوشه خلاصه کرد.

تلخیص ت

الگوریتم کامل برای نمایه‌سازی جریان داده پیوسته بر اساس پردازش داده‌های جریانی با بافرهایی اندازه ثابت و ادغام مداوم آنها در داده‌ساختار

تلخیص ت

الگوریتم ۱۱- اندیس کردن جریان داده با تلخیص ت

ورودی: جریان داده $D = \{x_1, x_2, \dots\}$

ورودی: اندازه بافر b، پارامتر فشرده‌سازی $\sigma > 1$، تابع مقیاس k

ورودی: داده‌ساختار t-digest

$T - DIGEST \leftarrow \emptyset$

تازمانی که D انجام بده

$B \leftarrow \{(x_1, 1), (x_2, 1), \dots, (x_b, 1)\}$

$T - DIGEST \leftarrow Merge(T - DIGEST, B, \sigma, k)$

برگرداندن T-DIGEST

تلخیص ت

تقسیم هزینه‌های زمان اجرای الگوریتم بافر-و-ادغام ۱۱ بین درج‌های مکرر مقادیر ورودی در بافر و فراخوانی‌های نادر الگوریتم ۱۰

درج‌ها ارزان

- هزینه‌های کلی تحت سلطه مرتب‌سازی و فراخوانی‌های تابع مقیاس در زیرالگوریتم ادغام
- سرشکنی در چندین درج

مثال - نمایه کردن جریان داده با تلخیص ت

مجموعه داده با $n = 20$

$\{0,0,3,4,1,6,0,5,2,0,3,3,2,3,0,2,5,0,3,1\}$

پارامتر فشرده‌سازی $\sigma = 5$، اندازه بافر $b = 10$ و تابع مقیاس مطابق تعریف قبلی

$$k(q,\sigma) = \frac{\sigma}{2\pi} arcsin\,(2q - 1)$$

مثال - نمایه کردن جریان داده با تلخیص ت

پر کردن بافر B با ده مقدار اول از ورودی

$$B = \langle (0,1),(0,1),(3,1),(4,1),(1,1),(6,1),(0,1),(5,1),(2,1),(0,1) \rangle$$

- با توجه به الگوریتم ۱۱، نیاز به ترکیب مقادیر بافر و مراکز جرم موجود در داده‌ساختار
- داده‌ساختار t-digest تهی: فهرست نامزدهای مراکز جرمX فقط شامل $n = 10$ مقدار از B
  - مرتب‌سازی صعودی آنها

$$X = \langle (0,1),(0,1),(0,1),(0,1),(1,1),(2,1),(3,1),(4,1),(5,1),(6,1) \rangle,$$

انتخاب اولین خوشه نامزد با چپ‌ترین مرکز جرم (۰,۱)

- محاسبهٔ مقدار مرزی چندک آن q_{limit}

$$q_{limit} = \frac{1}{2}\left[1 + sin\left(arcsin\,(-1) + \frac{2\pi}{5}\right)\right] = 0.34549$$

مثال - نمایه کردن جریان داده با تلخیص ت

دریافت مقدار بعدی از X

- دوباره (۰,۱)
- ارزیابی امکان ادغام آن بدون نقض ویژگی تلخیص در خوشه نامزد
- برای تخمین حداکثر مقدار چندک $\hat{q}$ این خوشه ادغام شده نیاز به

$$\hat{q} = \frac{1+1}{10} = 0.2,$$

- کمتر از مقدار مرزی چندک q_{limit}
  - امکان ادغام مقدار (۰,۱) در خوشه نامزد فعلی
  - اکنون دارای $c^{count} = 2$ مقدار
  - به دلیل یکسانی مقادیر، ثابت باقی ماندن مرکز جرم ثابت $c = 0$

به طور مشابه، امکان ادغام مقدار بعدی (۰,۱) در خوشه نامزد

- صرفا تغییر تعداد مقادیر خلاصه شده به $c^{count} = 3$

مثال - نمایه کردن جریان داده با تلخیص ت

دریافت مقدار چهارم از X

- دوباره (۰,۱)
- ارزیابی امکان ادغام آن بدون نقض ویژگی تلخیص در خوشه نامزد
- برای تخمین حداکثر مقدار چندک $\hat{q}$ این خوشه ادغام شده نیاز به

$$\hat{q} = \frac{3 + 1}{10} = 0.4,$$

- بیشتر از مقدار مرزی $q_{limit} = 0.34549$
  - عدم امکان ادغام مقدار (۰,۱) در خوشه نامزد فعلی
  - توقف تلاش برای جذب سایر مراکز جرم در خوشه
  - ذخیرهٔ آن در داده‌ساختار

$$T - DIGEST = \langle (0,3) \rangle,$$

- نگه‌داری حداکثر مقدار چندک آن به عنوان $q = \frac{c^{count}}{n} = 0.3$

مثال - نمایه کردن جریان داده با تلخیص ت

شروع به ایجاد خوشه نامزد جدید از مقدار در انتظار فعلی (۰,۱)، با $c = 0$ و $c^{count} = 1$، و مقدار مرزی چندک آن

$$q_{limit} = \frac{1}{2}\left[1 + sin\left(arcsin\,(2 \times 0.3 - 1) + \frac{2\pi}{5}\right)\right] = 0.874025$$

دریافت مقدار (۱,۱)

- در صورت ادغام آن در خوشه نامزد فعلی، حداکثر چندک تخمینی $\hat{q}$

$$\hat{q} = 0.3 + \frac{1+1}{10} = 0.5,$$

- عدم تجاوز از مقدار مرزی فعلی
- ادغام مقدار (۱,۱) در خوشه فعلی ادغام
- افزایش تعداد آن به $c^{count} = 2$ و بدل به مرکز جرم

$$c = 0 + \frac{1-0}{2} = 0.5$$

مثال - نمایه کردن جریان داده با تلخیص ت

به طور مشابه، پردازش بقیه مقادیر از X پردازش می‌کنیم و رشد داده‌ساختار به صورت زیر

$$T - DIGEST = \langle (0,3),(2,5),(5,1),(6,1) \rangle$$

با ادامه پردازش مجموعه داده، ایجاد بافر جدید پر

$$B = \langle (3,1),(3,1),(2,1),(3,1),(0,1),(2,1),(5,1),(0,1),(3,1),(1,1) \rangle,$$

پیوند آن با داده‌ساختار

- مرتب‌سازی صعودی بر اساس مراکز جرم دنباله حاصل X

$$X = \langle (0,3),(0,1),(0,1),(1,1),(2,5),(2,1),(2,1),(3,1),(3,1),(3,1),(3,1),(5,1),(5,1),(6,1) \rangle$$

تخیله داده‌ساختار

- آغاز از چپ‌ترین مقادیر و ادغام مقادیر به طور متوالی و ذخیره خوشه‌های در داده‌ساختار
- ادامه تا عدم امکان ادغام بیشتر

مثال - نمایه کردن جریان داده با تلخیص ت

داده‌ساختار حاصل دارای $m = 5$ خوشه:

$$T - DIGEST = \langle (0.1667,6),(2.36364,11),(5,1),(5,1),(6,1) \rangle$$

تلخیص ت-پرسش چندک

به دلیل از دست دادن اطلاعات دقیق در مورد مقادیر خوشه‌بندی شده

- به جز خوشه‌های تکی، که در آن مرکز جرم مقدار اولیه
- داده‌ساختار نمایشی با اتلاف از جریان داده
- نیاز به درون‌یابی برای تخمین چندک‌ها و پاسخ به پرس‌وجوی چندک،
  - با در نظر گرفتن توزیع خوشه‌هایی که توسط تابع مقیاس انتخاب شده تولید شده‌اند

تلخیص ت

الگوریتم ۱۲- پاسخ به پرس‌وجوهای چندک با تلخیص ت

ورودی: داده‌ساختار t-digest با m خوشه

ورودی: مقدار $q \in [۰,۱]$

خروجی: q - چندک

$n \leftarrow \sum_{j=۱}^{m} c_j^{count}$

اگر $n \cdot q < ۱$ آنگاه

برگرداندن $c_۱$

اگر $n \cdot q > n - \frac{۱}{۲} c_m^{count}$ آنگاه

برگرداندن c_m

// یافتن i به طوری‌که $\exists\, i \in [۱, m): q(c_i) + \frac{۱}{۲n} c_{i+۱}^{count} > q$.

// ⟸ چندک جستجو شده جایی بین c_i و $c_{i+۱}$

اگر $c_i^{count} == ۱$ و $q > q(c_i)$ آنگاه

برگرداندن c_i

اگر $c_{i+۱}^{count} == ۱$ و $q(c_{i+۱}) + \frac{۱}{n} \leq q$ آنگاه

برگرداندن $c_{i+۱}$

$\Delta_{\text{چپ}} \leftarrow (c_i^{count} == ۱)\,?\,۱ : ۰$

$\Delta_{\text{راست}} \leftarrow (c_{i+۱}^{count} == ۱)\,?\,۱ : ۰$

$w_{\text{چپ}} \leftarrow n \cdot q - n \cdot q(c_i) + \frac{c_i^{count} - \Delta_{\text{چپ}}}{۲}$

$w_{\text{راست}} \leftarrow n \cdot q(c_{i+۱}) - n \cdot q + \frac{c_{i+۱}^{count} - \Delta_{\text{راست}}}{۲}$

برگرداندن $\frac{c_i \cdot w_{\text{راست}} + c_{i+۱} \cdot w_{\text{چپ}}}{w_{\text{چپ}} + w_{\text{راست}}}$

تلخیص ت- پرسش چندک

یافتن q - چندک

- محاسبهٔ رتبه مقدار جستجو شده x در این دنباله مرتب شده
  - $n \cdot q$
- n تعداد کل مقادیر خلاصه شده در داده‌ساختار
- اگر رتبه کمتر از یک گزارش مرکز جرم c_1 به عنوان چندک
- اگر رتبه در نیمه آخرین خوشه آخر از حداکثر تعداد یا حتی بالاتر، گزارش c_m به عنوان حداکثر مقدار در تلخیص

تلخیص ت- پرسش چندک

در غیر این صورت،

- یافتن خوشه‌های C_i و C_{i+1} با مقادیر چندک تخمینی با معادله احاطه کننده مقدار چندک داده شده q

اگر خوشه چپ C_i تکی و حداکثر چندک آن بزرگتر از q

- برگرداندن مرکز جرم c_i به عنوان چندک
- مقدار خلاصه‌شده در خوشه

به طور مشابه، اگر خوشه راست C_{i+1} تکی و مقدار چندک حداقل تخمینی آن کوچکتر از q

- این خوشه مسئول چندک جستجو شده
- گزارش مرکز جرم c_{i+1} به عنوان بهترین تخمین برای آن

در غیر این صورت،

- محاسبهٔ وزن‌ها با ارزیابی سهم هر یک از این خوشه‌ها
- محاسبهٔ میانگین وزنی مراکز جرم از هر دو خوشه،
- ایجاد درون‌یابی جهت گزارش به عنوان q - چندک

تلخیص ت- پرسش چندک

وابستگی الگوریتم تخمین چندک به انتخاب تابع مقیاس

با توابع «توزیع‌شده‌تر؟» تولیدکنندهٔ دم بزرگتری از خوشه‌های تکی در دو سر طیف

- بهبود دقت برای چندک‌های دو سر طیف

علاوه بر این،

- توصیه بر ذخیرهٔ حداقل و حداکثر مقادیر در طول نمایه‌سازی و استفاده در درون‌یابی

مثال - پرس‌وجوی چندک با تلخیص ت

پرس‌وجوی چندک برای محاسبه چندک ۰٫۶۵ از داده‌ساختار حاصل قبلی

$$T - DIGEST = \langle (0.1667,6),(2.36364,11),(5,1),(5,1),(6,1) \rangle$$

مثال – پرس‌وجوی چندک با تلخیص ت

تعداد کل مقادیر مجموع شمارش‌های همه خوشه‌ها

- در مورد مثال $n = 20$
- رتبه چندک جستجو شده x برابر با $n \cdot q = 20 \cdot 0.65 = 13$
- نه کمتر از یک و نه خیلی نزدیک به n،
- جستجو برای دو خوشه متوالی C_i و C_{i+1}
  - مراکز جرم آنها احاطهٔ کنندهٔ چندک جستجو شده

در داده‌ساختار فعلی، برابر C_2 و C_3

- زیرا

$$q(c_2) + \frac{1}{2\times 20} c_3^{count} = \frac{6+11}{20} + \frac{1}{40} = 0.875 > 0.65$$

مثال - پرس‌وجوی چندک با تلخیص ت

خوشه C_3 تکی

- $c_3^{count} = 1$
- بررسی حداکثر مقدار چندک آن با مقدار چندک جستجو شده q
  - جهت تعیین بهترین تناسب بودن مرکز جرم آن

$$q(c_3) + \frac{1}{20} = \frac{6 + 11 + 1}{20} + \frac{1}{20} = 0.95,$$

به طور قابل توجهی برزگتر از مقدار $q = 0.65$

چندک واقعی در جایی بین مراکز جرم c_3 و c_2

- اما خوشه راست با تنظیم $\Delta_{right} = 1$ تکی

مثال – پرس‌وجوی چندک با تلخیص ت

بنابراین، تخمین چندک جستجو شده با درون‌یابی با استفاده از میانگین وزنی مراکز جرم

- $w_{\text{چپ}} = 20 \cdot 0.65 - 20 \cdot q(c_2) + \frac{c_2^{count} - 0}{2} = 13 - 20 \cdot \frac{6 + 11}{20} + \frac{11}{2} = 1.5,$
- $w_{\text{راست}} = 20 \cdot q(c_3) - 20 \cdot 0.65 + \frac{c_3^{count} - 1}{2} = 20 \cdot \frac{6 + 11+1}{20} - 13 + \frac{1 - 1}{2} = 5$

مثال – پرس‌وجوی چندک با تلخیص ت

در نهایت، چندک ۰٫۶۵ تخمینی برای مجموعه داده

$$x = \frac{c_2 \cdot w_{\text{راست}} + c_3 \cdot w_{\text{چپ}}}{w_{\text{چپ}} + w_{\text{راست}}} = \frac{2.36364 \times 5 + 5 \times 1.5}{1.5 + 5} = 2.97$$

- بسیار نزدیک به مقدار دقیق ۳ برای آن مجموعه داده

پرس‌وجوی چندک معکوس

یافتن رتبه مقدار داده شده x

مقایسه مقدار x با حداقل و حداکثر مراکز جرم در داده‌ساختار

- به ترتیب چپ‌ترین و راست‌ترین خوشه‌ها

اگر x خارج از آن بازه

- بسته به سمت ظهور مقدار x خروجی: یک یا تعداد کل مقادیر n برابر مقدار رتبه تخمینی

در غیر این صورت، به دنبال مقدار x در میان مراکز جرم

- در صورت یافتن چنین خوشه‌هایی
  - جمع کردن شمارنده‌های آنها
  - گزارش رتبه (x) به عنوان رتبه خوشه با کوچکترین نمایه با مقدار تنظیم شده

پرس‌وجوی چندک معکوس

ناموفق بودن بررسی‌های قبل

- اطمینان از قرار گرفتن مقدار x بین مراکز جرم خوشه‌های متوالی، مثلاً (c_i,c_{i+1}) قرار می‌گیرد و رتبه آن حداقل $n \cdot q(c_i)$
- اگر هر دو خوشه تکی، به این معنی که مراکز جرم آنها دقیقاً مقادیر ورودی
  - عدم نیاز به به تصحیح آن مقدار و برگرداندن آن به عنوان رتبه جستجو شده
- زمانی که فقط یکی از آن خوشه‌ها تکی باشد، رتبه را با سهم مقیاس‌گذاری شده خوشه دیگر تنظیم می‌کنیم تا مقدار نهایی را به دست آوریم.
- در غیر این صورت، با استفاده از اندازه هر دو خوشه، ایجاد درون‌یابی و محاسبهٔ مقدار رتبه

الگوریتم ۱۲- پاسخ به پرس‌وجوهای چندک معکوس با t-digest

ورودی: مقدار x

ورودی: داده‌ساختار t-digest با m خوشه

خروجی: رتبه عنصر

$n \leftarrow \sum_{j=۱}^{m} c_j^{count}$

اگر $x < c_۱$ آنگاه برگرداندن ۱

اگر $x > c_m$ آنگاه برگرداندن n

// بررسی مرکز جرم بودن x

اگر $\exists\, j: c_j == x$ آنگاه

$J \leftarrow \{j: c_j == x\}, i^* \leftarrow min\,(J)$

برگرداندن $n \cdot q(c_i^*) - c_{i^*}^{count} + \frac{۱}{۲}\sum_{j \in J} c_j^{count}$

// در این مرحله، می‌توانیم مطمئن باشیم که $\exists\, i \in [۱, m): x \in (c_i, c_{i+۱})$

رتبه $\leftarrow n \cdot q(c_i)$

اگر $c_i^{count} == ۱$ و $c_{i+۱}^{count} == ۱$ آنگاه

برگرداندن رتبه

$\hat{x} \leftarrow \frac{x - c_i}{c_{i+۱} - c_i}$

اگر $c_i^{count} == ۱$ آنگاه

برگرداندن رتبه $+ \frac{\hat{x}}{۲} \cdot c_{i+۱}^{count}$

اگر $c_{i+۱}^{count} == ۱$ آنگاه

برگرداندن رتبه $- \frac{۱ - \hat{x}}{۲} \cdot c_i^{count}$

برگرداندن رتبه $+ \frac{\hat{x}}{۲} \cdot c_{i+۱}^{count} - \frac{۱ - \hat{x}}{۲} \cdot c_i^{count}$

پرس‌وجوی بازه

شبیه تلخیص چ

- انجام دو پرس‌وجوی چندک معکوس
- یافتن تفاوت بین رتبه‌های مرزهای بازه

ویژگی‌ها

سبک-سنگینی بین اندازه داده‌ساختار(کنترل شونده با پارامتر فشرده‌سازی σ) و سرعت و دقت تخمین چندک‌ها

امکان دستیابی به سرعت بیشتر با استفاده از مقدار ثابت حافظه

- با انتخاب مقدار کوچکتر σ و اندازه بافر بزرگ b

بالاترین دقت،

- ترجیح استفاده از σ بزرگتر برای فشرده‌سازی کمتر و بافر بزرگتر (به عنوان مثال، $\sigma \times 10$)

کوچکترین حافظه

- ترجیح استفاده از بافر کوچکتر و مقادیر بزرگتر پارامتر فشرده‌سازی σ

ویژگی‌ها

بازهٔ تعداد خوشه‌ها (m) در داده‌ساختار و با نمایه‌سازی $n \geq \frac{\sigma}{2}$ مقدار و برآورده‌سازی ویژگی تلخیص

$$\left\lfloor \frac{\sigma}{2} \right\rfloor \leq m \leq \lfloor \sigma \rfloor$$

مثال - تخمین فضای مورد نیاز

در پی نمایه‌سازی حداقل $n = 1000$ مقدار با پارامتر فشرده‌سازی $\sigma = 100$

- انتظار ۵۰ تا ۱۰۰ خوشه کاملاً ادغام شده

در داده‌ساختار

- نمایش هر خوشه با مرکز جرم و تعداد مقادیر نمایه شده
- با داشتن شمارنده‌های ۳۲ بیتی و عدد ممیز شناور با دقت دو برابر ۶۴ بیتی برای مقدار مرکز جرم
  - کل مرکز جرم نیازمند ۱۲ بایت حافظه
  - جا شدن کل داده‌ساختار در حدود ۱٫۲ کیلوبایت حافظه

مثال – تخمین فضای مورد نیاز

برای دقت بالا،

- استفاده از بافری ده برابر بزرگتر از پارامترهای فشرده‌سازی
- با داشتن $b = 10 \cdot \sigma = 1000$
  - تخصیص شمارنده‌های ۱۶ بیتی کوچکتر و اعداد ممیز شناور ۶۴ بیتی
  - در زمان اجرا به ۱۰ کیلوبایت حافظه اضافی

ویژگی‌ها

دستیابی به دقت ϵ در تخمین q - چندک
- متناسب با $q \cdot (1 - q)$

سایر الگوریتم‌ها صرفا در پی کنترل خطای مطلق ثابت

تلخیص ت در پی کنترل خطای نسبی
- مقاوم کردن الگوریتم در برابر خطاهای قابل توجه برای چندک‌های انتهای طیف

مزیت تلخیص ت به تلخیص چ
- مدیریت مقادیر ممیز شناور در مقابل محدود بودن به اعداد صحیح

ویژگی‌ها

ادغام دو داده‌ساختار تلخیص ت با استفاده از الگوریتم ادغام

- عدم پیوستگی داده‌ساختار حاصل
- اما تولید تخمین خوبی بر اساس نتایج تجربی

امکان موازی‌سازی برای بخش‌های مختلف جریان داده

- استفاده در پاسخ به پرس‌وجوهای رتبه از آنها
- مفید در عملیات MapReduce و استخراج جریان برای برنامه‌های داده بزرگ