الگوریتم
معرفی

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

قدیم و جدید

الگوریتم- ۱۴۰۴-۱۴۰۳ فصل زمستان

- تمرکز
  - تحلیل زمانی و مرتبهٔ رشد
  - تقسیم‌وحل
  - الگوریتم‌های حریصانه
  - برنامه‌ریزی پویا
  - اشاره‌ای مقدماتی به P و NP و پیچیدگی رایانشی
- پیام اصلی
  - طراحی الگوریتم‌های دقیق و بهینه

تحلیل زمان بدترین زمان اجرا

مهارپذیری مسائل در ردهٔ P

مدل کلاسیک نظریهٔ الگوریتم‌ها

قدیم و جدید

الگوریتم- ۱۴۰۵-۱۴۰۴ فصل بهار

- تغییر قواعد با داده‌های عظیم
- محیطی متفاوت
  - داده‌های عظیم
  - جریان داده (streaming)
  - یک‌بار عبور (single-pass)
  - محدودیت شدید حافظه
  - پردازش آنلاین
  - نیاز به زمان و حافظهٔ زیرخطی
- تفاوت اساسی
  - روش‌های دقیق کلاسیک O(n)
  - اجراناپذیری برای مقیاس میلیاردی

قدیم و جدید

الگوریتم- ۱۴۰۵-۱۴۰۴ فصل بهار

- شاهد
  - شمارش کاربران یکتا به‌صورت دقیق ← $O(n)$ حافظه ← ناممکن برای $n = 10^{11}$
  - نگهداری همهٔ اقلام برای بررسی دقیق عضویت ← غیرعملی
  - شمارش فراوانی‌ها با لغت‌نامه ← بسیار حجیم
  - جستجوی دقیق شباهت ← $O(n^2)$ ← غیرقابل اجرا
- سخن کوتاه
  - معیار مهارپذیری نبودن چندجمله‌ای به دلیل بزرگی n

تعریف جدید مهارپذیری

دور قبل

- حل‌پذیر: در زمان چندجمله‌ای

دور کنون

- حل‌پذیر: در حافظهٔ کم + یک‌بار پیمایش یا گذر + زمان زیرخطی

تغییری در تعریف کلید فهم حاضرات

- امکان بودن مسئله در ردهٔ P
  - اما حل‌ناپذیری در مقیاس داده‌های عظیم
  -

ابزارهای جدید- داده‌ساختار و الگوریتم‌های احتمالی

استفاده از ابزارهای جدید جهت دستیابی به درهم‌سازی

- درهم‌سازی (hashing)
- MinHash
- LSH
- Bloom Filter
- Count-Min Sketch
- HyperLogLog
- Reservoir Sampling
- Bottom-k Sampling
- تصادفی‌سازی
- تحلیل احتمال و تمرکز measure-ها

ابزارهای جدید- داده‌ساختار و الگوریتم‌های احتمالی

ابزارهای جدید فراهم‌ساز

- عضویت
- شمارش بسامد
- تخمین تعداد یکتا
- جستجوی تشابه
- اقلام شایع یا پربسامد heavy hitters

اما با حل در

- حافظهٔ ثابت یا بسیار کم
- زمان O(1) یا O(log n)
- حالت جریانی
- با خطای کنترل‌شده

ما برای وصل آمدیم

در ایام ماضی تدریس درس حاضر در پی سریع‌ترین الگوریتم دقیق برای حل مسئله
- تلقی مهارپذیری مسائل کلاسیک در P

عدم کاربرد تعریف مذکور در دنیای داده‌های عظیم
- حتی عملی نبودن الگوریتم‌های O(n)، حتی جدول درهم دقیق، حتی P روی دادهٔ کامل، با n میلیارد

در فصل جاری تعریفی جدید
- الگوریتم‌های زیرخطی و داده‌ساختارهای احتمالی.

در عوض توجه به دقت کامل،
- پذیرش خطای کوچک و قابل‌کنترل
- قابل قبول بودن سرعت و مصرف حافظهٔ

سخن کوتاه این درس ادامه‌ای طبیعی بر درس سال گذشته است،
- صرفا تغییر در محیط محاسباتی

از کجا آمده‌ام وآمدنم بهر چه بود؟!

ایام ماضی:

- تحلیل بدترین حالت
- صحت ۱۰۰ درصد
- زمان چندجمله‌ای
- مدل‌های قطعی

مذهب مختار:

- صحت تقریبی با تضمین
- تحلیل امیدریاضی و با احتمال بالا
- زمان و حافظهٔ زیرخطی
- تصادفی‌سازی
- سبک-سنگینی بین خطا، سرعت، و حافظه

این تغییر، «مدل ذهنی جدید» دانشجوست.

از کجا آمده‌ام وآمدنم بهر چه بود؟!

عدم حذف نظریهٔ الگوریتم‌ها --- گسترش می‌یابد

بودن الگوریتم‌های احتمالی بر پایهٔ همان اصول تحلیل، صرفا تعویض ابزارها

- تحلیل خطای فیلتر بلوم: احتمال و درهم‌سازی
- تحلیل دقت شمارش کمینه: تصادم درهم‌ساز
- HyperLogLog یا ابرلگ‌لگ: آمار و نظریهٔ تخمین
- LSH: احتمال برخورد در درهم‌سازی
- نمونه‌برداری: قانون اعداد بزرگ و تمرکز

سخن کوتاه، این درس ادامهٔ منطقی است، نه موضوعی جدید.

مثال- شمارش تعداد کاربران یکتا

روش‌های معمول:

- مجموعه درهم دقیق: فضا O(n)

روش فعلی:

- ابرلگ‌لگ تقریبی: فضا چند کیلوبایت O(1)

مثال- عضویت

روش‌های معمول:

- جدول درهم دقیق: حافظهٔ زیاد O(n)

روش فعلی:

- فیلتر بلوم: تقریبی: چند بایت O(k)

سبک‌سنگینی دقت در مقابل کارائی

مثال- جستجوی شباهت

مقایسهٔ دو سند

- مقایسهٔ دو سند ⟶ نیاز به بررسی همهٔ کلمات
- مقایسهٔ دو بردار ⟶ نیاز به محاسبهٔ کل فاصله
- مقایسهٔ دو مجموعه ⟶ نیاز به بررسی همهٔ عناصر
- مقایسهٔ دو تصویر ⟶ نیاز به مقایسهٔ تعداد زیادی پیکسل

پرهزینه

- O(n) یا O(d) یا حتی $O(n^2)$ در مقایسه‌های دو به دو

مزیت درهم‌سازی

مثال- جستجوی شباهت

تبدیل اشیای بزرگ را به اثرانگشت‌های کوچک با درهم‌سازی

استفاده از درهم دو شی به جای مقایسهٔ آنها

- h(A)
- h(B)

هر درهم معادل عدد متناظر یا امضای کوتاه است.

صرفا مقایسهٔ اثرانگشت‌ها به جای مقایسهٔ اشیای بزرگ

سرعت بیشتر چنین بهبودی

مثال- جستجوی شباهت

تبدیل مقایسهٔ کامل به مقایسهٔ چند عدد با درهم‌سازی

مثال: دو مجموعهٔ بزرگ

$$S1 = \{a1, a2, \dots, a1000000\}$$

$$S2 = \{b1, b2, \dots, b1000000\}$$

نیاز به بررسی اقلام فراوان در صورت مقایسهٔ مستقیم

محاسبهٔ MinHash

- minhash(S1) = 284
- minhash(S2) = 284

درک سریع شباهت دو مجموعه با روش مزبور

- مقایسهٔ یک عدد به جای میلیون‌ها عنصر

مثال- جستجوی شباهت

حذف مقایسه‌های جفتی با LSH

فرض وجود N سند و در پی یافتن اسناد مشابه

- روش جستجوی کامل
  - مقایسهٔ جفت‌ها
  - تعداد مقایسه‌ها: N^2
  - غیرعملی بودن برای داده‌های عظیم

اما با Locality Sensitive Hashing (LSH):

- محاسبهٔ یک امضا برای هر سند
- قرارگیری امضاها را در سطل با استفاده درهم
- مقایسه صرف اسناد واقع در یک سطل
- درهم‌ساز به مثابهٔ فیلتر اولیه

مثال- جستجوی شباهت

کاهش بعد داده با درهم‌سازی بعد داده

در جستجوی شباهت برداری:

- بردار ممکن است ۷۶۸ یا ۱۰۲۴ بعد داشته باشد.
- محاسبهٔ فاصله بین دو بردار یعنی صدها ضرب و جمع.

استفاده از درهم‌سازهایی چون SimHash یا LSH

- نگاشت بردارها به سطل‌ها
- نتیجه: به جای مقایسهٔ همهٔ بردارها با هم، فقط بردارهایی را بررسی می‌کنیم که در سطل مشابه هستند.

مثال- جستجوی شباهت

ایجاد میانبر احتمالی با درهم‌سازی

- به جای اینکه هر شیء را با همهٔ اشیاء دیگر مقایسه کنیم،
- درهمسازی هر شیء به طوری که:
  - قرارگیری اشیای مشابه در یک سطل با احتمال زیاد
  - قرارگیری اشیای نامشابه در سطل‌های مختلف

صرفا بررسی تصادم درهم‌سازی

سخن کوتاه

- گران بودن مقایسه دو شی بزرگ اما ارزان بودن درهم انها
- در صورت امکان طراحی تابع درهم با تولید مقدار مشابه برای اشیا مشابه: امکان استفاده جهت مقایسهٔ کامل

مثال- جستجوی شباهت

به دنبال یافتن اسناد مشابه در ۱۰۰۰ سند

- روش مستقیم: ۱۰۰۰ × ۱۰۰۰ یا یک میلیون مقایسه.
- روش LSH:
  - تبدیل هر سند به یک امضای MinHash با ۵۰ عدد
  - قرار دادن امضاها در سطل‌ها
  - هر سطل شامل محتملا پنج تا ده سند
  - فقط همان‌ها را مقایسه می‌کنیم
  - در نتیجه امکان ۵۰۰۰ یا ۱۰۰۰۰ مقایسه به جای ۱,۰۰۰,۰۰۰

درهم‌سازی موجب کاهش بسیار زیاد محاسبات

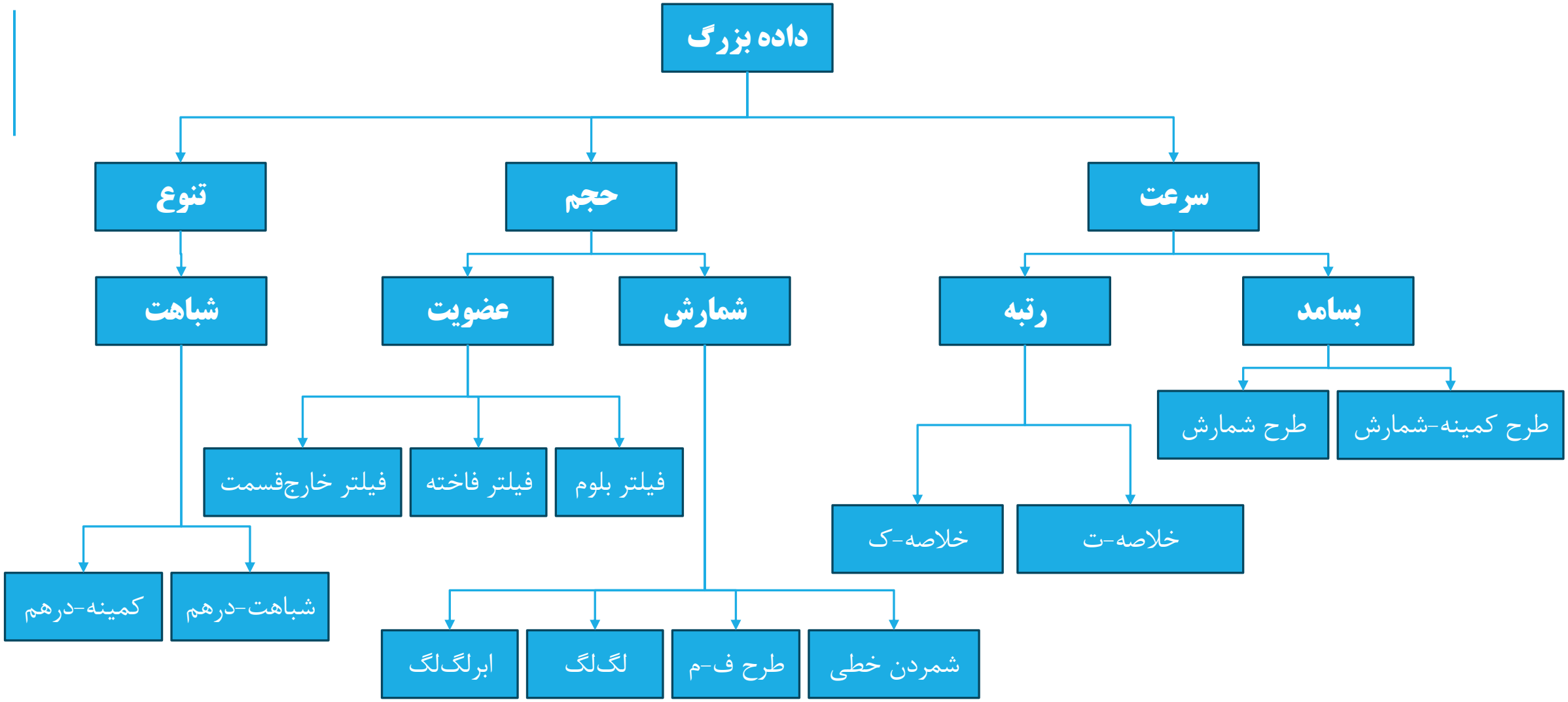
مثال- جستجوی شباهت

روش‌های معمول:

- شباهت دقیق: مقایسه جفت دقیق $O(n^2)$

روش فعلی:

- MinHash + LSH: زمان زیرخطی


داده بزرگ
سرعت
حجم
تنوع
بسامد
رتبه
عضویت
شمارش
شباهت
طرح کمینه-شمارش
طرح شمارش
خلاصه-ت
خلاصه-ک
فیلتر بلوم
فیلتر فاخته
فیلتر خارج‌قسمت
شباهت-درهم
کمینه-درهم
شمردن خطی
طرح ف-م
لگ‌لگ
ابرلگ‌لگ

سخن کوتاه

یادگیری طراحی الگوریتم‌های دقیق در سال قبل

یادگیری طراحی الگوریتم‌های تقریبی، احتمالی و زیرخطی قابل‌اجرا روی مقیاس‌های بزرگ

بارم

تمرین- نوشتاری و برنامه‌نویسی همراه معرفی دقیق و علمی مراجع

تحقیق (مطالعه و معرفی یا نوآوری) ده تا پانزده صفحه به فارسی فونت ۱۱ دو ستونه با مراجع کافی و مناسب و مرتبط

امتحان؟

تصحیح متون

- سطح لغوی
- سطح جمله
- سطح مفهومی
- پیشنهادات برای تکمیل

موضوعات تحقیق

موضوعات پیشنهادی با به صورت مطالعه (سروی) یا بهبود (مهلت انتخاب تا ۱۵ اردیبهشت در گروه‌های دو نفره)

- Learned Data Structures
- بهبودها و کاربردهای فیلتر بلوم فراتر از درس (مانند *Learned Bloom Filters (LBF)*)
- بهبودهای دیگر و کاربردهای ابرلگ‌لگ (مثلا Sliding Window، HLL++ ، یا HLL for network flows)
- الگوریتم‌های موازی و توزیع‌شده احتمالی
- Streaming Algorithms for Sliding Windows بررسی الگوریتم‌هائی مانند *WVHLL* یا *Datar-Gionis-Indyk-Motwani (DGIM)*
- الگوریتم‌های دیگر جهت کار با داده بزرگ
- Fine-grained complexity
- الگوریتم‌های متأخر مسئلهٔ کوله‌پشتی
- الگوریتم‌های متاخر ضرب ماتریسی و «بهبود» آن
- آزمون اول بودن

منابع درس

[Medjedovic22] D, Medjedovic, E. Tahirovic, “Algorithms and Data Structures for Massive Datasets,” Simon and Schuster, 2022.

[Gakhov20] A. Gakhov, Probabilistic Data Structures And Algorithms For Big Data Applications, BoD, 2020.

[Rocca21] M. La Rocca, Advanced Algorithms and Data Structures, Simon and Schuster, 2021.

[Neapolitan]

[CLRS]

[Anand2011] J. Leskovec, A. Rajaraman, J.D. Ullman, Mining of Massive Datasets. Cambridge university press, 2020.