دانشگاه شهید باهنر کرمان

صفر: معرفی

ساختمان داده ها و الگوریتم

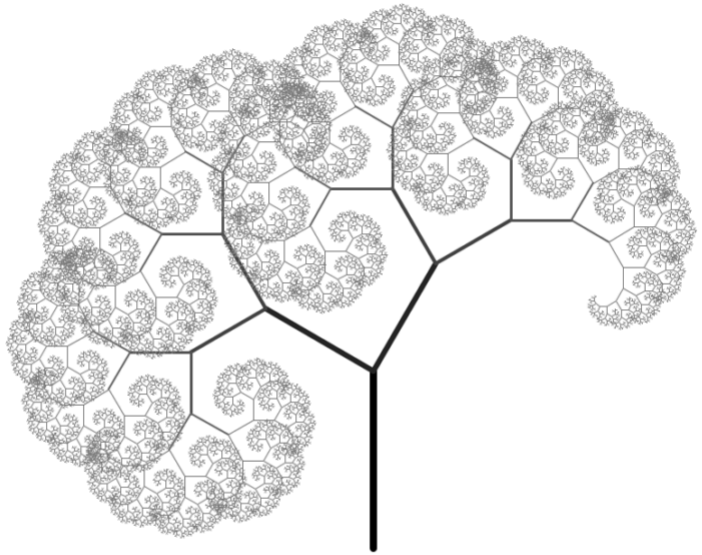
مدرس: دکتر نجمه منصوری

نگارنده: سجاد هاشمیان

Barnsley Fern Fractal

این درس درباره چیست؟

- حل مسئله و برنامه نویسی به همراه مثال های کاربردی
- الگوریتم، روشی به منظور حل مسئله
- ساختمان داده، روشی به منظور ذخیره اطلاعات



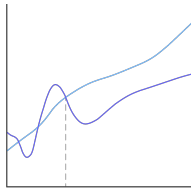
اهداف درس

- تبدیل یک برنامه‌نویس مبتدی به یک برنامه‌نویس خوب
  - آشنایی با ساختمان داده های متداول
  - آشنایی با روشهای مختلف پیاده‌سازی
  - مقایسه کارایی روش‌های مختلف پیاده‌سازی
- بررسی برخی از کاربردها

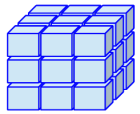
به نظر من، تفاوت میان یک برنامه‌نویس بد و خوب در این است که برنامه‌نویسان بد بیشتر به کد اهمیت می‌دهند در حالی که برای برنامه‌نویس‌های خوب ساختمان داده‌ها اهمیت بیشتری دارد.

لینوس تروالدز (مبدع لینوکس)

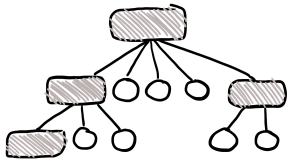
سرفصل ها

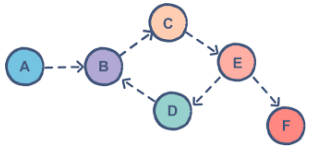


- تحلیل الگوریتم ها
- ساختمان داده‌های ابتدایی :
  - آرایه، پشته، صف، لیست پیوندی

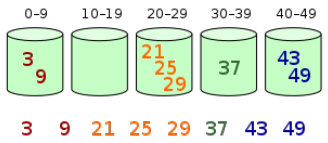


- درخت ها :
  - هرم، درخت دودویی

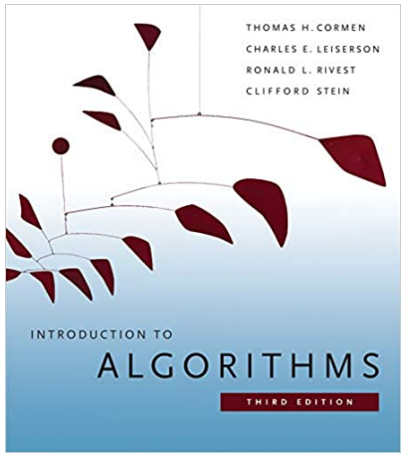


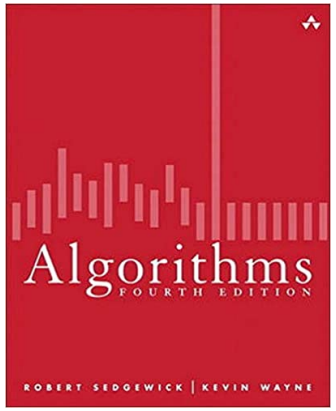


- گراف ها :
  - نمایش گراف، پیمایش گراف(جستجوی عمقی و سطحی)
  - یافتن کوتاه‌ترین مسیر، محاسبه درخت پوشای کمینه
- مرتب‌سازی :
  - حبابی، درجی، انتخابی، ادغامی، درختی، هرمی، سریع، مبنایی



منابع

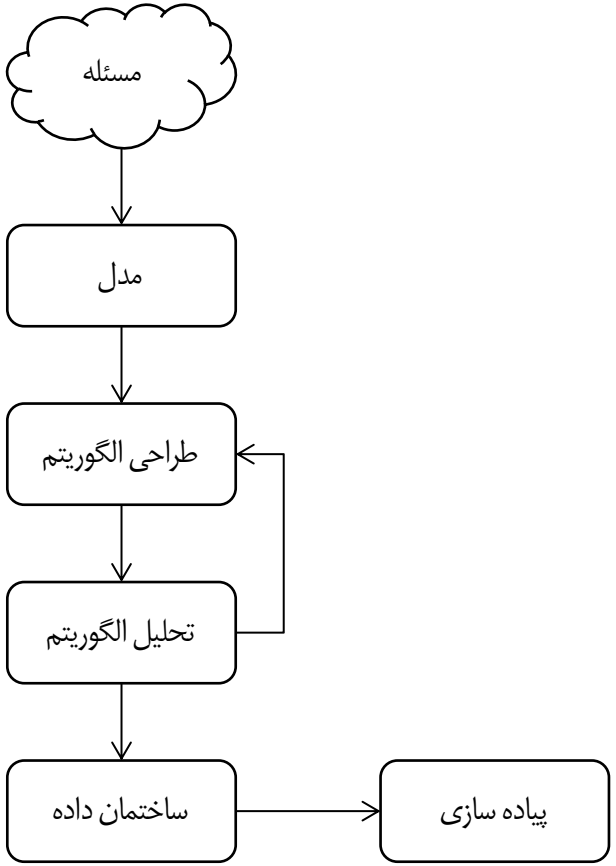




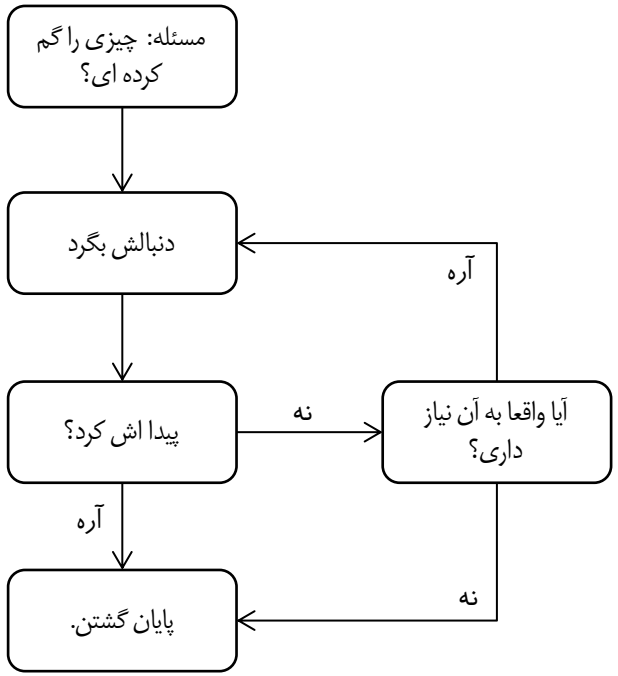
- ساختمان داده‌ها و طراحی الگوریتم
  - ایهو، هاپکرافت، آلمن
- مقدمه ای بر الگوریتم ها
  - کورمن، لایسرسون، ریوست، اشتاین
- الگوریتم ها
  - رابرت سژویک، کوین وین
- داده ساختار ها و مبانی الگوریتم ها
  - محمد قدسی

مراحل حل مسئله

- ایجاد یک مدل انتزاعی از مسئله – **مدل**
- طراحی الگوریتم برای حل مدل – **طراحی الگوریتم**
- تحلیل کارائی الگوریتم – **تحلیل الگوریتم**
- تعریف ساختمان‌داده های لازم – **ساختمان داده**
- پیاده‌سازی الگوریتم ارائه شده – **پیاده سازی**
- ...



خصوصیات الگوریتم



- **ورودی:** حداقل "۰" ورودی (ممکن است ورودی نداشته باشد)
- **خروجی:** حداقل "۱" خروجی (در غیر این صورت، چرا باید اجرا شود؟)
- **قطعیت:** هرکدام از دستور ها باید دقیق و بدون ابهام باشند.
- **محدودیت:** هر الگوریتم باید پس از طی مراحل مشخصی پایان پذیرد.
- **کارائی:** امکان اجرا و پیاده سازی برای آن وجود داشته باشد، یعنی انجام شدنی باشد.

چالش

- آیا برنامه من می‌تواند یک مسئله با اندازه ورودی بزرگ را حل کند؟

مثال: ۳-مجموع

با داشتن آرایه ای از N عدد صحیح متمایز، چند سه تایی وجود دارد که مجموع آنها دقیقا برابر با ۰ شود؟

```
public static int count(int N,int[] a)
{
    int count=0
    for(int i=0;i<N;i++)
       for(int j=i+1;j<N;j++)
           for(int k=j+1;k<N;k++)
             if(a[i]+a[j]+a[k]==0)
                 count++;
    return count;
}
```

الگوریتم کورکورانه: بررسی تمام ۳-تایی ها

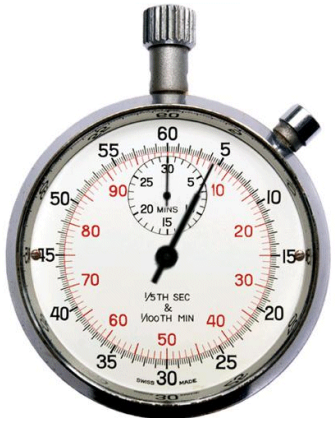
a[i]	a[j]	a[k]	sum
3	-4	1	0
3	-2	-1	0
-4	4	0	0
-1	0	1	0

```
N: 8
Array: [3, -4, -2, -1, 4, 0, 1, 5]
Answer: 4
```

اندازه گیری زمان اجرا

سوال) چگونه زمان اجرای یک برنامه را اندازه بگیریم؟

پاسخ: به صورت دستی!؟



اندازه گیری زمان اجرا

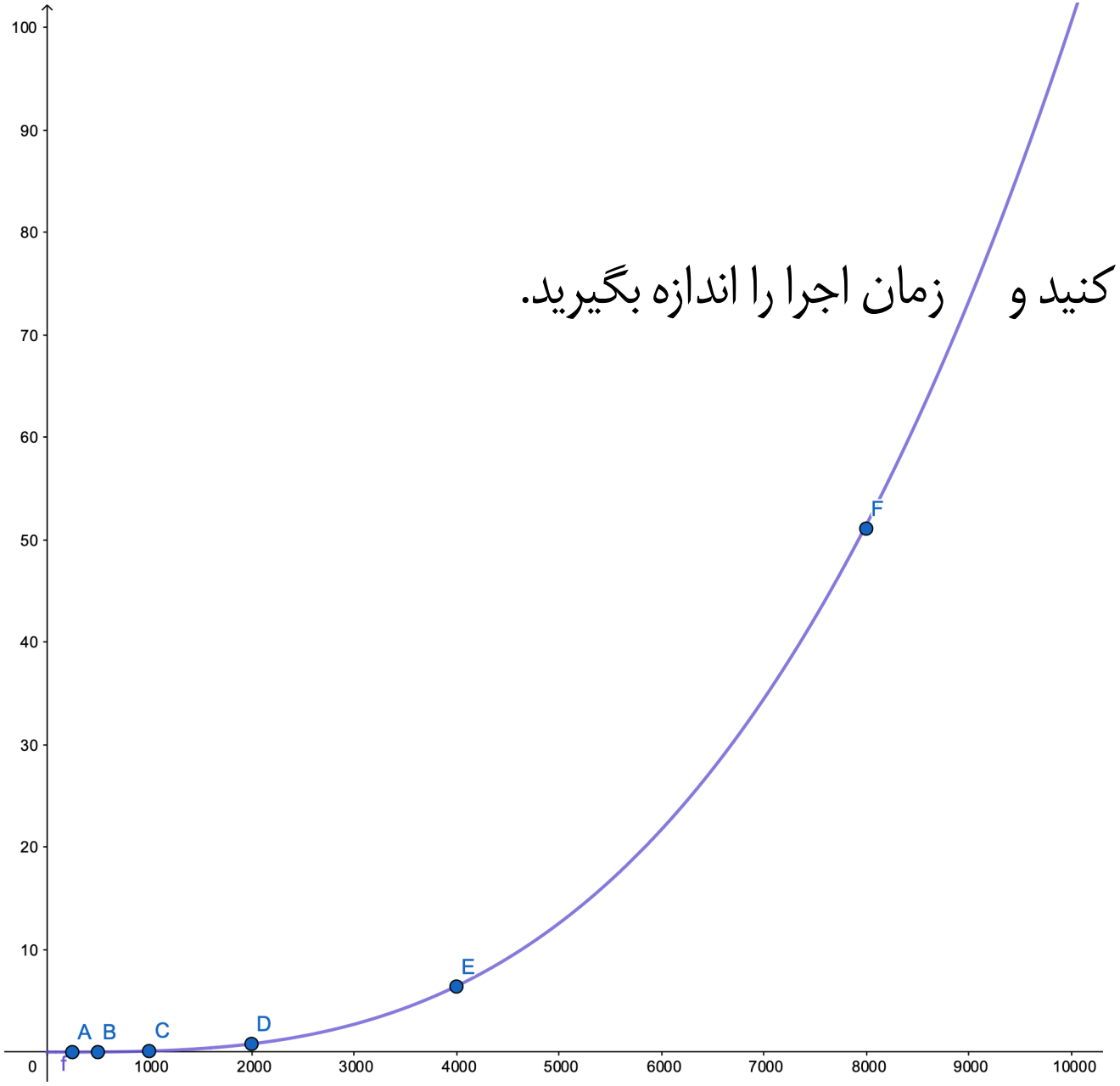
سوال) چگونه زمان اجرای یک برنامه را اندازه بگیریم؟

پاسخ: به صورت خودکار

```
public static void main(String[] args)
{
    int[] a = In.readInts(args[0]);
    Stopwatch timer = new Stopwatch();
    StdOut.println(ThreeSum.count(a));
    double time = timer.elapsedTime();
}
```

تحلیل تجربی

- برنامه را به ازای اندازه های مختلف ورودی اجرا کنید و زمان اجرا را اندازه بگیرید.



N	Time(second)
250	0.0
500	0.0
1000	0.1
2000	0.8
4000	6.4
8000	51.1
16000	?

پیش‌بینی و اعتبار سنجی

- فرضیه. زمان اجرا تقریبا برابر با $۱/۰۰۶\times۱۰^{-۱۰}\times N^{۲.۹۹۹}$ ثانیه است.

- پیش‌بینی.
  - ۵۱ ثانیه برای N=۸۰۰۰
  - ۴۰۸/۱ ثانیه برای N=۱۶۰۰۰
- مشاهدات.
  - اجرای الگوریتم

N	Time(second)
250	0.0
500	0.0
1000	0.1
2000	0.8
4000	6.4
8000	51.1
16000	410.8

تحلیل پیچیدگی و مرتبه اجرایی

در ارزیابی یک الگوریتم و برنامه دو عامل اصلی وجود دارد که باید مورد توجه قرار گیرند، یکی حافظه مصرفی و دیگری زمان مصرفی است، یعنی الگوریتمی بهتر است که فضا و زمان کمتری را در لحظه اجرا مصرف کند.

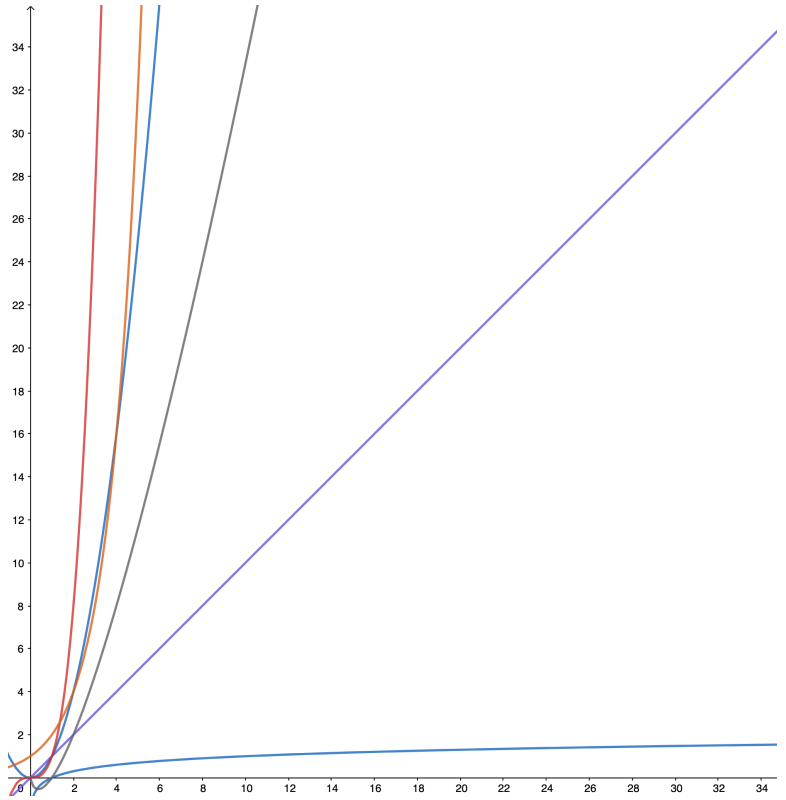
محاسبه زمان اجرایی یک الگوریتم متناسب با تعداد دفعاتی است که **عملیات اصلی** انجام می‌شود. در تحلیل زمان اجرای یک الگوریتم تمام دستورات را شمارش نمی‌کنیم زیرا تعداد دستورات به نوع زبان برنامه نویسی، کامپیوتر و ... بستگی دارد.

بنابراین برای محاسبه زمان اجرای یک الگوریتم فقط به عملیات اصلی نیاز داریم که مستقل از عوامل دیگر هستند؛ در عین حال هیچ قاعده مشخصی برای انتخاب عملیات اصلی وجود ندارد!

دسته بندی نرخ رشد الگوریتم ها

یک مجموعه‌ی کوچک از توابع مانند زیر برای توصیف نرخ رشد اغلب الگوریتم ها کافی است.

$1, \log N, N, N \log N, N^2, N^3, 2^N$



دسته بندی نرخ رشد الگوریتم ها

مرتبه رشد	نام	مثال
۱	ثابت	عملیات های پایه (جمع دو عدد)
log N	لگاریتمی	جستجو دودویی (با ما باشید)
N	خطی	یافتن عنصر بیشینه
N log N	خطی لگاریتمی	مرتب سازی ادغامی (با ما باشید)
N^2	درجه دو	دو حلقه تو در تو
N^3	درجه ۳	سه حلقه تو در تو
2^N	نمایی	جستجو کامل(مسئله برج هانوی)

انواع تحلیل

- بهترین حالت. حد پایین هزینه
  - با پیدا کردن «ساده ترین» ورودی
  - فراهم کننده یک هدف کمینه برای تمام ورودی ها
- بدترین حالت. حد بالا هزینه
  - با پیدا کردن «سخت ترین» ورودی
  - فراهم کننده یک تضمین بیشینه برای تمام ورودی ها
- حالت متوسط. هزینه مورد انتظار برای یک ورودی تصادفی
  - نیاز به یک مدل برای ورودی «تصادفی»
  - فراهم کننده روشی برای تخمین کارایی

مثال ۳-مجموع: تعداد عملیات ها در الگوریتم کورکورانه:
بهترین حالت: $\sim 0.5 \times N^3$
حالت متوسط: $\sim 0.5 \times N^3$
بدترین حالت: $\sim 0.5 \times N^3$