



مباحث پیشرفته

نگاهی دوباره به الگوریتم‌های مرتب‌سازی

مرتب‌سازی توسط الگوریتم‌های مختلفی انجام می‌شود که برای آنها، تقسیم‌بندی‌های مختلفی وجود دارد:

- درونی / بیرونی
- پایدار / ناپایدار
- نوع ساختمان داده (آرایه / لیست پیوندی)
- مقایسه‌ای / غیرمقایسه‌ای
-

پوشش: برای هر یک از انواع روش‌های مرتب‌سازی که در بالا آمده است، مثالی ارائه کنید. 

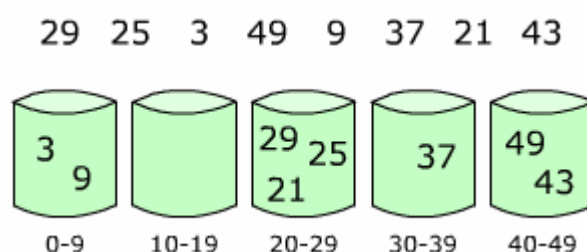
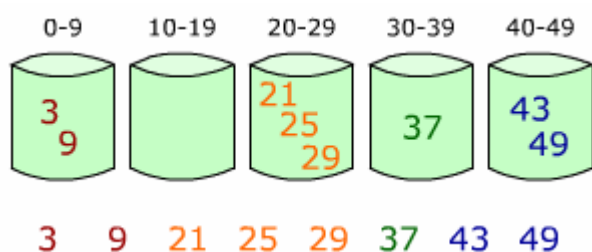
نکته: مرتب‌سازی مقایسه‌ای از نظر مرتبه پیچیدگی به دو نوع کلی مرتبه $\theta(n^2)$ و مرتبه $\theta(n \lg n)$ تقسیم می‌شود. 


مرتب‌سازی غیرمقایسه‌ای

دسته‌ای از روش‌های مرتب‌سازی وجود دارند که عمل مرتب‌سازی را با روش‌هایی نظیر دسته‌بندی و شاخص‌گذاری (ایندکس‌گذاری) انجام می‌دهند نظیر:

- مرتب‌سازی سطلی
- مرتب‌سازی مبنایی
- مرتب‌سازی شمارشی
- مرتب‌سازی لانه کبوتری
-

مرتب‌سازی سطلی (Bucket Sort)



پوشش: کارآیی مرتب‌سازی سطلی به چه عواملی بستگی دارد؟ 


روند کلی مرتب‌سازی سطلی

- 1) Set up an array of initially empty "buckets."
- 2) **Scatter**: Go over the original array, putting each object in its bucket.
- 3) Sort each non-empty bucket.
- 4) **Gather**: Visit the buckets in order and put all elements back into the original array.

شبه کد الگوریتم مرتب‌سازی سطلی

```
function bucket-sort(array, n) is
  buckets ← new array of n empty lists
  for i = 0 to (length(array)-1) do
    insert array[i] into buckets[msbits(array[i], k)]
  for i = 0 to n - 1 do
    next-sort(buckets[i])
  return the concatenation of buckets[0], ..., buckets[n-1]
```

Alg. 10-1

پرسش: مرتبه پیچیدگی مرتب‌سازی سطلی چیست؟ 

مرتب‌سازی مبنایی (Radix Sort)

From right to left

239 234 879 878 123 358 416 317 137 225

			123	234	225	416	317 137	878 358	239 879
--	--	--	-----	-----	-----	-----	---------	---------	---------

	416 317	123 225	234 137 239		358		878 879		
--	---------	---------	-------------	--	-----	--	---------	--	--


	123 137	225 234 239	317 358	416			878 879		
--	---------	-------------	---------	-----	--	--	---------	--	--

شکل ۱۰-۳: مراحل الگوریتم مرتب‌سازی مبنایی روی یک مثال

مثالی دیگر

170, 45, 75, 90, 2, 24, 802, 66

1. sorting by least significant digit (1s place) gives:
170, 90, 2, 802, 24, 45, 75, 66
2. sorting by next digit (10s place) gives:
2, 802, 24, 45, 66, 170, 75, 90
3. sorting by most significant digit (100s place) gives:
2, 24, 45, 66, 75, 90, 170, 802

پرسش: با توجه به عملکرد مرتب‌سازی سطلی و مرتب‌سازی مبنایی، چگونه می‌توان مرتب‌سازی مبنایی را مشابه مرتب‌سازی سطلی دانست؟
به عبارت دیگر چگونه می‌توان عملکرد مرتب‌سازی مبنایی را مبتنی بر روش مرتب‌سازی سطلی بیان کرد؟ 

آشنایی با نظریه NP

تا کنون با الگوریتم‌های مختلفی برای مساله فروشنده دوره گرد (TSP) آشنا شده‌اید، به طور ضمنی این نتیجه تداعی می‌شود که پیچیدگی زمانی این الگوریتم‌ها در بدترین حالت نمایی است؛ با این وجود، هنوز کسی ثابت نکرده است که طراحی الگوریتم‌های بهتر (با مرتبه پیچیدگی کمتر) امکان‌پذیر نیست. این گونه مسائل، مباحث و نظریه‌های مرتبط به آن در حوزه نظریه NP است.

تعریف و توضیح چند اصطلاح مهم

الگوریتم زمانی چند جمله‌ای: الگوریتمی که پیچیدگی زمانی آن در بدترین حالت در حد بالا تابع چند جمله‌ای بر حسب اندازه ورودی باشد یعنی؛ $W(n) \in O(p(n))$ که $p(n)$ چند جمله‌ای است.

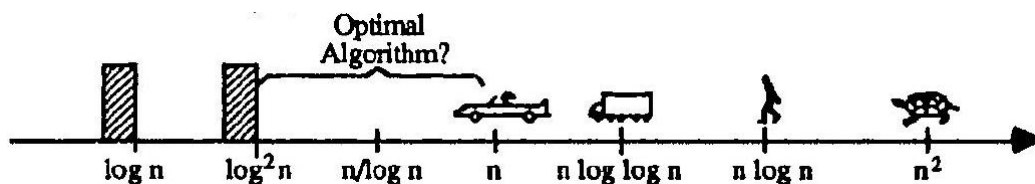
مسائل بغرنج (Intractable): به طور ساده به مسائلی گفته می‌شود که ارائه یک الگوریتم کارآمد برای آنها غیرممکن است. مساله بغرنج مساله‌ای است که حل آن توسط یک الگوریتم زمانی چند جمله‌ای غیرممکن باشد. (بغرنج بودن از ویژگی‌های مساله است نه الگوریتم)

مسائل از نظر بغرنج بودن به سه گروه تقسیم می‌شوند:

- ۱- مسائلی که الگوریتم چند جمله‌ای برای آنها یافت شده است.
- ۲- مسائلی که بغرنج بودن آنها ثابت شده است.
- ۳- مسائلی که بغرنج بودن آنها ثابت نشده است ولی هیچ الگوریتم زمانی چند جمله‌ای هم برای آنها یافت نشده است.

پرسش: برای هر یک از مسائل سه گروه بالا، مثال یا مثال‌هایی ارائه کنید؟

نکته: در مطالب پیشین به کلاس‌ها یا دسته‌های مختلف پیچیدگی الگوریتم‌ها، اشاره شد. در کاربردهای واقعی، به دلیل بزرگ بودن اندازه ورودی الگوریتم‌ها، معمولاً مرتبه پیچیدگی الگوریتم‌ها باید کمتر یا مساوی مرتبه دوم باشد.



شکل ۱۰-۴: مقایسه الگوریتم‌ها از نظر مرتبه پیچیدگی در کاربردهای واقعی

مساله تصمیم‌گیری (Decision Problem): مسائلی که خروجی آنها، بله یا خیر است.

مثال: مساله تصمیم‌گیری فروشنده دوره گرد: آیا به ازای عدد مثبت d توری با وزن کل کوچکتر یا مساوی d وجود دارد یا خیر؟

پرسش: مساله کوله‌پشتی را به صورت یک مساله تصمیم‌گیری بیان کنید.

مسائل کلاس P: مسائلی تصمیم‌گیری که توسط الگوریتم‌های چند جمله‌ای قابل حل (Solvable) هستند.

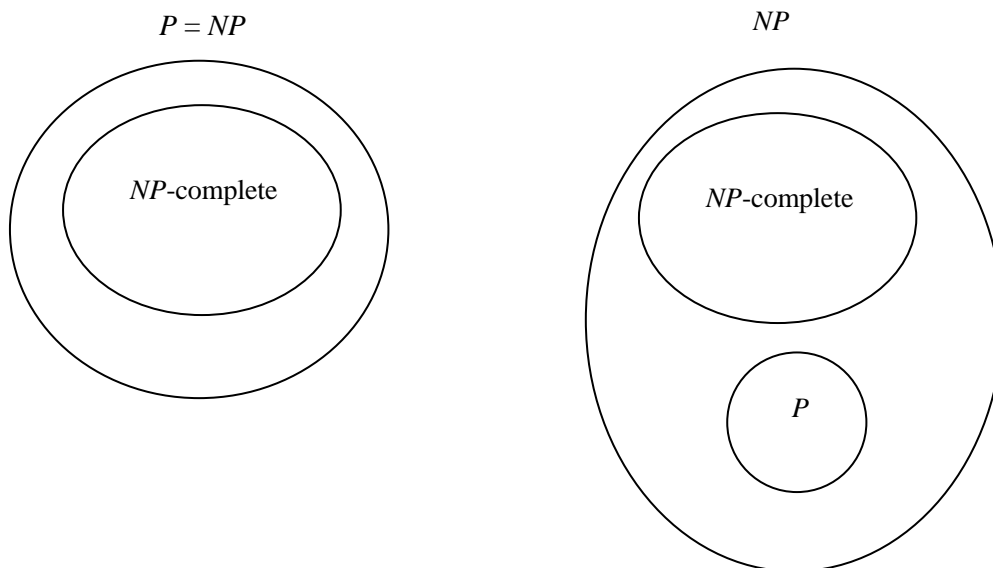
مثال: تعیین وجود یک کلید در یک آرایه

مسائل کلاس NP: مسائل تصمیم‌گیری که توسط الگوریتم‌های غیرقطعی زمانی چندجمله‌ای قابل حل هستند. به عبارت دیگر، اگر پاسخ مساله را داشته باشیم، می‌توانیم در زمان چندجمله‌ای تعیین کنیم که آیا پاسخ، متعلق به آن مساله است یا خیر. هر الگوریتم غیرقطعی دو مرحله مهم دارد:

۱- مرحله حدس (غیرقطعی): تولید یک راه حل مثل یک رشته S که ممکن است این رشته بی ربط باشد.

۲- مرحله تصدیق (قطعی): آیا راه حل متعلق به مساله است یا نه؟

مسائل کلاس NP-Complete: بخش خاصی از مسائل NP هستند با این ویژگی که اگر بتوان مساله‌ای از این مجموعه را در زمان چندجمله‌ای حل کنیم، می‌توانیم تمام مسائل NP را هم در زمان چندجمله‌ای حل کنیم.



شکل ۱۰-۵: رابطه بین مسائل کلاس P و کلاس NP

پرسش: با توجه به مفهوم مسائل کلاس NP چه باید کرد؟ آیا باید منتظر بود تا یکی از مسائل NP-Complete در زمان چندجمله‌ای حل شود؟

آشنایی با الگوریتم‌های موازی

یکی از روش‌های تسریع در اجرای الگوریتم‌ها، طراحی آنها به صورت موازی است. یعنی الگوریتم بتواند بر روی چند پردازنده به صورت موازی و همزمان اجرا شود.

مثال: محاسبه مجموع ۸ عدد به صورت موازی با چهار پردازنده

تمرین‌ها:

تمرین ۱۰-۱: تحقیق کنید روش‌های مرتب‌سازی شمارشی و لانه کبوتری چگونه عمل می‌کنند. مرتبه پیچیدگی زمانی آنها چیست؟

تمرین ۱۰-۲: بررسی کنید که چگونه آلن تورینگ نشان داده‌است که مساله توقف (Halting Problem)، غیرقابل تصمیم‌گیری است.

تمرین ۱۰-۳: در کنار مسائل NP-Complete، مسائلی به نام NP-Hard وجود دارند. تحقیق کنید که مسائل NP-Hard چه نوع مسائلی هستند.

تمرین ۱۰-۴: تحقیق کنید طراحی الگوریتم‌های موازی با چه چالش‌هایی روبروست؟

مسائلی که به دلیل سطح فعلی تفکر ما بوجود می‌آیند، نمی‌توانند با همان سطح تفکر، حل گردند. (آلبرت اینشتین)