



ساختمان داده‌ها و الگوریتم‌ها

نیم‌سال اول ۱۴۰۰-۰۱
مدرس: مسعود صدیقین

یادآوری جلسه هجدهم

مرتب سازی

در جلسه قبل، راجع به روش‌های مختلف مرتب سازی صحبت کردیم. انواع روش‌های مرتب سازی را از نظر زمان و همینطور معیارهای زیر مورد بررسی قرار دادیم:

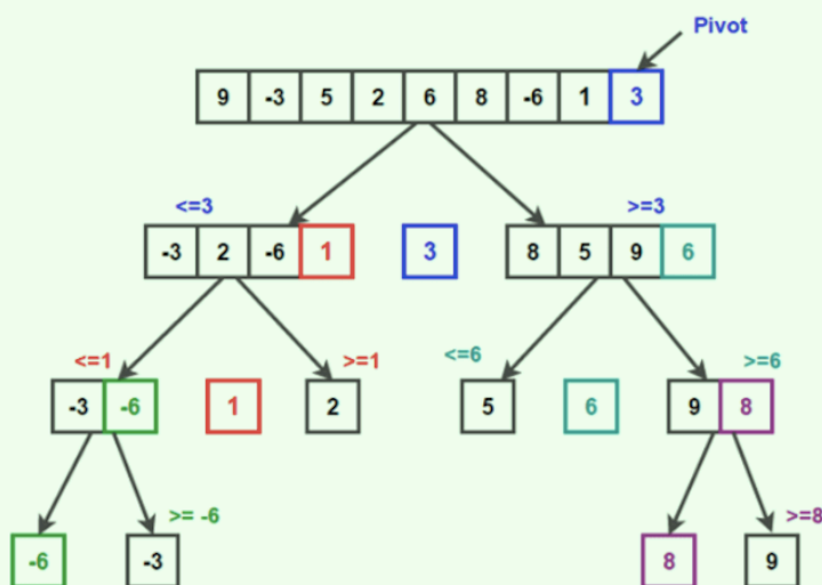
- پایدار بودن: ترتیب عناصر برابر را بعد از مرتب سازی حفظ می‌کند.
- درجا بودن: به غیر از آرایه اعداد، از حافظه دیگری استفاده نکند.

همچنین به معرفی یک مرتب سازی جدید به نام مرتب سازی سریع پرداختیم. برای معرفی این مرتب سازی، ابتدا یک زیر مساله به نام *partition* را مورد مطالعه قرار دادیم. در این مساله، یک آرایه شامل اعداد و یک اندیس x داده شده است. هدف، قرار دادن عدد مربوط به اندیس x در محل صحیح آن (در حالت مرتب شده) و همچنین قرار دادن اعداد کوچکتر در سمت چپ و اعداد بزرگتر در سمت راست آن است. دقت کنید که لازم نیست سمت راست و چپ لزوماً مرتب باشند. نشان دادیم که عمل *partition* را می‌توان در زمان $O(n)$ انجام داد.

بر این اساس، مرتب سازی سریع به این صورت است: (فرض کنید آرایه $[l, r]$ قرار است مرتب شود).

- آرایه را بر حسب عنصر دلخواه x پارتیشن کنید. فرض کنید p محل این عنصر بعد از پارتیشن باشد.
- دو بخش $[l, p-1]$ و $[p+1, r]$ را به طور بازگشتی توسط مرتب سازی سریع مرتب کنید.

شکل زیر، مرتب سازی سریع یک آرایه را نشان می‌دهد. دقت کنید که در این مثال، همیشه سمت راست ترین عنصر به عنوان محور انتخاب شده است و بر اساس آن *partition* انجام می‌شود.



مرتب سازی سریع تصادفی، یک مرتب سازی است که در آن، عنصر انتخاب شده به عنوان محور یک عنصر تصادفی است. در جلسه قبل نشان دادیم که متوسط زمان اجرای مرتب سازی تصادفی $O(n \log n)$ است.

تمرین: مرتب‌سازی سریع را سه بار مرور کنید. در ابتدای این جلسه، هدف حل این مساله است: اگر عنصر محور به صورت تصادفی انتخاب شود، در این صورت متوسط زمان اجرای مرتب‌سازی تصادفی چقدر خواهد بود؟

