



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

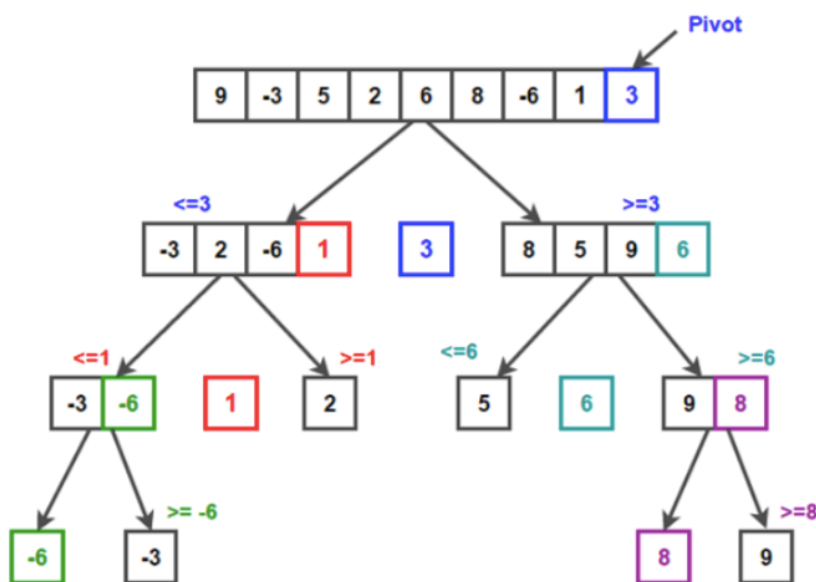
- پایدار بودن: ترتیب عناصر برابر را بعد از مرتب‌سازی حفظ می‌کند.
- درجا بودن: به غیر از آرایه اعداد، از حافظه دیگری استفاده نکند.

همچنین به معرفی یک مرتب‌سازی جدید به نام مرتب‌سازی سریع پرداختیم. برای معرفی این مرتب‌سازی، ابتدا یک زیرمساله 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 مرتب‌شده) 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، مرتب‌سازی 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 مرتب‌سازی سریع مرتب کنید.

شکل زیر، مرتب‌سازی 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.



۱. مرتب‌سازی حبابی

۲. مرتب‌سازی درجی

۳. مرتب‌سازی ادغامی

۴. مرتب‌سازی هرمی

۵. مرتب‌سازی با استفاده از دج متوازن

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

