

به نام خدا



درس طراحی الگوریتم

---

تمرین سری سوم

---

مدرس درس:

سرکار خانم دکتر ملکی

تهیه شده توسط:

الناز رضایی ۹۸۴۱۱۳۸۷

تاریخ ارسال: ۱۴۰۱/۰۱/۱۴

## سوال ۱:

آرایه‌ای با سایز  $n$  ورودی می‌گیریم. می‌خواهیم با کمترین حرکت ممکن اعداد زوج و فرد را توسط عوض کردن جای دو همسایه از هم جدا کنیم. تعداد جابه‌جایی‌ها را بشمارید.

۱. روش حل مسئله را ارائه دهید و پیچیدگی زمانی و حافظه‌ای آن را بیان دهید.

۲. چرایی حریصانه بودن راه‌حل خود را توضیح دهید (بهینگی آن را اثبات کنید).

## پاسخ ۱.۱:

برای حل این سوال، از روش greedy استفاده می‌کنیم؛ به این صورت که مسئله را به دو زیرمسئله تقسیم می‌کنیم و یک بار تعداد جابه‌جایی‌های لازم برای آوردن اعداد فرد در سمت چپ آرایه و یک بار تعداد جابه‌جایی‌ها برای قرار گرفتن اعداد فرد در سمت راست آرایه را محاسبه کرده و مینیمم این دو را به عنوان خروجی می‌دهیم.

در ابتدا دو متغیر  $left$  و  $right$  به ترتیب برای ذخیره کردن تعداد جابه‌جایی‌های لازم برای انتقال اعداد فرد به سمت چپ و راست را با مقدار صفر تعریف می‌کنیم. سپس یک بار آرایه را از ابتدا به انتها پیمایش کرده و هر بار که به عددی فرد رسیدیم، متغیر  $left$  را با تعداد مراحلی که باید انجام دهیم تا آن عدد به مکان مناسبی در آرایه جابجا شود جمع می‌کنیم. حال بار دیگر آرایه را از انتها به ابتدا پیمایش کرده و تعداد جابه‌جایی‌های لازم برای انتقال عناصر فرد به انتهای آرایه را در متغیر  $right$  می‌ریزیم. سپس مینیمم مقدار  $left$  و  $right$  را محاسبه کرده و به عنوان خروجی نمایش می‌دهیم. پیچیدگی زمانی چون حلقه‌های  $for$  به صورت مجزا و حاوی  $n$  بار هستند، از  $O(n)$  order می‌باشد و پیچیدگی فضایی چون آرایه نداریم و فقط چند متغیر است، از  $O(1)$  order می‌باشد.

## پاسخ ۲.۱:

در این الگوریتم، علاوه بر اینکه حالات تکراری در نظر گرفته نمی‌شود، آرایه هم استفاده نشده تا مقادیر اضافی ذخیره شوند و فضای بیهوده اشغال شود. به علاوه دلیل حریصانه بودن این روش این است که در هر مرحله، فقط دو عنصر اول آرایه را بررسی می‌کنیم و در صورت لزوم جابه‌جایی انجام می‌دهیم، بدون این که سایر عناصر را بررسی کنیم. به عبارت دیگر، مرتب‌سازی انجام شده در هر

مرحله فقط برای دو عنصر اول است و سایر عناصر تغییری نمی‌کنند. این باعث می‌شود که این روش با پیچیدگی زمانی  $O(n)$  یکی از سریع‌ترین الگوریتم‌های ممکن برای این مسئله باشد.