

طراحی الگوریتم – پاییز ۱۳۹۶

تکلیف برنامه نویسی نخست

مدرس: هلاکو رحمانیان

مقدمه

در این تکلیف برنامه نویسی به پیاده سازی راه حل مسئله ی Counting Inversions خواهیم پرداخت. به صورت خاص دو الگوریتم زیر را با رسم نمودار اندازه – زمان با هم مقایسه خواهیم کرد:

۱. الگوریتم کورکورانه: در این الگوریتم تمام زوج ها را بررسی کرده و تعدادهای وارون ها را می شماریم. همانطور که می دانید مرتبه ی زمانی این الگوریتم مرتبه ی $O(n^2)$ است یا همان

$$O(n^2)$$

۲. الگوریتم تقسیم و غلبه: در این الگوریتم آرایه را به دو زیر آرایه می شکنیم و تعداد وارون ها را در هر دو زیر آرایه و بین زیر آرایه ها شمرده و با هم جمع می کنیم. می دانیم که مرتبه زمانی این الگوریتم عبارتست از

$$O(n \log n)$$

قالب و الگوی کد

قالب کدی که در اختیار شما قرار داده شده است از سه فایل تشکیل شده است:

۱. HW1.java

این فایل فایلی اجرایی است. با اجرای این فایل می توانید از درستی توابعی که نوشته اید اطمینان حاصل کرده و در صورت درستی توابع داده های لازم برای ترسیم نمودار را ترسیم کنید. این فایل تکمیل است و شما نیازی به تغییر آن ندارید.

۲. InversionCount.java

این فایل حاوی الگوریتم های شماست. این فایل سه تابع خالی دارد که شما می بایست آن را پر کنید.

۳. Coordinator.java

این فایل شامل کد لازم را برای تولید داده برای ترسیم نمودار است. این فایل نیز تکمیل است و شما نیازی به تغییر آن ندارید.

بخش هایی که شما باید پُر کنید

قبل از هر چیز توجه داشته باشید که با اینکه شما تنها لازم است بخشی از کد را کامل کنید اما می بایست تمام کد در فایل های مختلف را بفهمید. به ترتیب توابع ذیل را از فایل InversionCount.java پر کنید:

الف) bruteForce: این تابع حاوی الگوریتم کورکورانه خواهد بود. یک آرایه به عنوان ورودی دریافت کرده و به عنوان خروجی یک عدد طبیعی برمی گرداند.

ب) mergeAndCount: این تابع بخش ادغام الگوریتم تقسیم و غلبه را انجام می دهد. این تابع دو آرایه نخست را ترکیب کرده و در آرایه ی سوم قرار می دهد و همچنین به عنوان خروجی تعداد وارون های میان دو زیر آرایه را تحت یک عدد طبیعی برمی گرداند.

sortAndCount: این تابع حاوی الگوریتم تقسیم و غلبه خواهد بود. یک آرایه به عنوان ورودی دریافت کرده و به عنوان خروجی یک عدد طبیعی برمی گرداند. این تابع آرایه را به دو زیرآرایه ی (تقریباً) مساوی می شکند، تعداد واریون ها را در هر زیرآرایه به صورت بازگشتی می شمارد و نهایتاً دو آرایه را ترکیب کرده و با استفاده از تابع قبلی تعداد واریون های میان دو زیرآرایه را نیز شمرده و تعداد کل واریون ها را به عنوان حاصل جمع این سه در قالب خروجی برمی گرداند.

اجرا و تحلیل

اگر توابع را به درستی پر کنید، پس از اجرای فایل HW1.java یک فایل جدید تحت نام `plot_data.csv` تولید خواهد شد. این فایل را با نرم افزار Excel باز کنید. سه ستون از اعداد خواهید دید. این سه ستون را انتخاب کرده و آن ها را در قالب نمودار با دستور زیر ترسیم کنید:

Charts → Scatter

نمودار ترسیم شده به مانند تصویر زیر خواهد بود. همانطور که می بینید با بزرگ شدن اندازه ی آرایه، میزان زمانی لازم برای حل مسئله در الگوریتم کورکورانه به صورت سهموی (مربعی) رشد می کند؛ حال آنکه میزان رشد برای الگوریتم تقسیم و غلبه بسیار ناچیز است.

