



تمرین کامپیوتری سوم



سیستم‌های عامل - بهار 1400

گزارش کار

دانشکده مهندسی برق و کامپیوتر

نام و نام خانوادگی:

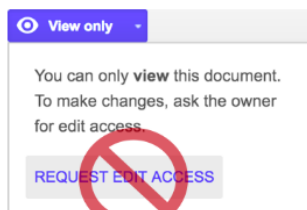
علیرضا توکلی

تاریخ:

۹ خرداد ۱۴۰۰

استاد:

دکتر مهدی کارگهی

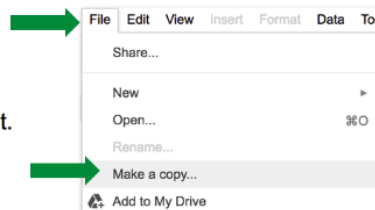


How to use this template:

This is a view-only file and cannot be edited.

Create your own copy of this template to edit.

In the menu, click **File > Make a copy...**



2

مقدمه

3

پیاده‌سازی سری

3

سوال اول

3

سوال دوم

3

جدول اول

3

پیاده‌سازی چندریسه‌ای

3

سوال سوم

4

سوال چهارم

4

سوال پنجم

4

جدول دوم

مقدمه



در این تمرین شما به تحلیل داده‌هایی که از مشخصات و قیمت فروش خانه‌ها جمع‌آوری شده‌است پرداخته شده است. در ابتدا برنامه اقدام به خواندن و تجزیه مجموعه داده¹ی ارائه شده کرده و آنها را در حافظه خود ذخیره می‌کند. پس از استخراج داده‌ها و ویژگی‌های آنها، برنامه اقدام به برچسب گذاری ستون قیمت داده‌ها، بدست آوردن میانگین و انحراف معیار داده‌ها و در نهایت اقدام به تعیین رده قیمتی خانه‌ها می‌کند. این تمرین به دو روش این مسئله پیاده‌سازی شده است که در ادامه گزارش، نتایج حاصل آمده است.



¹ Dataset

پیاده‌سازی سری

سوال اول

چرا برای پیاده‌سازی یک برنامه بصورت چندریشه‌ای، بهتر است ابتدا این برنامه بصورت سری پیاده‌سازی شود؟ همان‌طور که گفته شده، دیباگ کردن برنامه‌ی چندریشه‌ای کار بسیار سخت و زمان‌بری است. این کار باعث می‌شود از درست بودن تقریبی منطق اولیه‌ی برنامه اطمینان حاصل کرد. به علاوه با این کار می‌توان نقاط hotspot را نیز شناسایی کرد که این کار به ما کمک می‌کند تصمیم بهتری برای انتخاب چندریشه‌ای کردن برنامه بگیریم.

سوال دوم

با بررسی زمان اجرای بخش‌های مختلف برنامه، Hotspot² های برنامه را مشخص کنید. از آنجایی که برنامه تکه تکه و به صورت تابع نوشته شده است، زمان صرف شده در هر تابع را اندازه‌گیری می‌کنیم. خواندن دیتاست، ۰.۰۱۲۹ ثانیه
۰ و ۱ کردن ستون آخر، ۰.۰۰۰۱۱۰۰۰۶ ثانیه
محاسبه‌ی میانگین و واریانس، ۰.۰۰۰۱۵۲۳ ثانیه
پیشبینی دسته‌ی هر نمونه و چاپ کردن دقت برنامه، ۰.۰۰۰۳۲۴۱۰۳ ثانیه
به طول انجامید که همان‌طور که قابل مشاهده است، ۹۵ درصد برنامه صرف خواندن دیتا می‌شود.

جدول اول

زمان‌های اجرای ۶ اجرای متوالی از برنامه و میانگین آن‌ها را به ازای ورودی نمونه‌ای که در شرح تمرین آمده است، در جدول زیر بیاورید.

میانگین	اجرای ششم	اجرای پنجم	اجرای چهارم	اجرای سوم	اجرای دوم	اجرای اول
0.0104294	0.01488689	0.01545982	0.00513561	0.00523811	0.01140702	0.01044915

² توابعی که در برنامه‌تان بیشترین زمان اجراها را به خود اختصاص می‌دهند.

پیاده‌سازی چندریسه‌ای

سوال سوم

اگر هنگام موازی‌سازی برنامه به زمان اجرای بیشتری نسبت به حالت سری برخورد کنید، چه رویکردهایی را برای کاهش زمان اجرا و استفاده حداکثری از موازی‌سازی پیش می‌گیرید؟

(۱) تغییر دادن تعداد ریشه‌ها

(۲) بررسی ادامه‌ی کد و بررسی بخش‌های دیگر برنامه که قابل موازی‌سازی دارند

(۳) بررسی قسمت‌های موازی شده که تغییری در زمان اجرائی نداشته‌اند

سوال چهارم

در هنگام پیاده‌سازی این بخش، به چه چالش‌هایی برخورد کردید و بیان کنید که به چه صورت آن‌ها را رفع کردید.

(۱) حواسمان باید می‌بود که داده‌هایی که در اشتراک همه‌ی ریشه‌ها هستند، باید با mutex محافظت شوند وگرنه ممکن است اشکالی پیش آید. نحوه‌ی درست کردن این مشکل نیز با mutex است که در کدها مشخص است.

(۲) نحوه‌ی انتقال داده‌های مخصوص به ریشه به ریشه مشکل بود که با ساختن struct مربوطه این مشکل حل شد.

(۳) کدام بخش‌ها را نیازمند است که موازی‌سازی کنیم و کدام بخش‌ها صرفاً باعث اضافه شدن سریار به برنامه می‌شوند. راه حل هم برای این مشکل نیز به این صورت بود که با بررسی کدها هزینه‌ی تقریبی هر تکه را می‌توان فهمید. راه حل دیگر هم این بود که حالت‌های مختلف را بررسی کردم تا نتیجه مورد نظر حاصل شد.

سوال پنجم

با توجه به تجربه‌ای که در پیاده‌سازی این تمرین بدست آوردید، به نظر شما در چه مواقعی استفاده از مکانیزم‌های همگام‌سازی مانند قفل^۳ در یک طراحی چندریسه‌ای ضروری است؟ در این پروژه از چه مکانیزم‌هایی استفاده کرده اید؟ آیا استفاده از این مکانیزم‌ها می‌تواند اثر مخربی داشته باشد؟

استفاده از قفل زمانی نیاز است که امکان استفاده‌ی دو ریشه از یک حافظه وجود دارد. توجه شود که در صورتی که یک ریشه روی یک بازه از یک آرایه و ریشه‌ی دیگر روی بازه‌ی دیگری از آرایه در حال نوشتن و خواندن هستند، مشکلی وجود ندارد و نیازی نیست

^۳ Lock

از قفل استفاده کنیم. در این پروژه در زمانی که می‌خواستیم داده‌های هر بخش را به داده‌های اصلی اضافه کنیم، از mutex استفاده کردیم که باعث می‌شد از خراب‌کاری جلوگیری کند. توجه کنید که در صورتی که می‌توانستیم محاسبات را انجام دهیم، اول محاسبات را انجام می‌دادیم و در آخر و در ذخیره‌سازی داده‌ها تنها از mutex استفاده می‌کردیم تا زمان اجرایی کم شود. اگر استفاده از mutex بی‌رویه باشد، به دلیل قفل شدن دیگر ریشه‌ها، زمان اجرایی زیاد می‌شود که از اثرات بد این روش است. اما در شرایطی ناچار به استفاده از این مکانیزم خواهیم بود.

جدول دوم

زمان‌های اجرای ۶ اجرای متوالی از برنامه و میانگین آن‌ها را به ازای ورودی نمونه‌ای که در شرح تمرین آمده است، در جدول زیر بیاورید.

میانگین	اجرای ششم	اجرای پنجم	اجرای چهارم	اجرای سوم	اجرای دوم	اجرای اول
0.0062256	0.002908	0.010438	0.010360	0.003706	0.002677	0.007265

میزان تسریع $\left(\frac{Serial\ Time}{Parallel\ Time} \right)$ برنامه نسبت به حالت سری را در زیر بیاورید.

میزان تسریع	میانگین زمان اجرای موازی	میانگین زمان اجرای سری
1.675226214	0.0062256	0.0104294