

تمرین کامپیوتری سوم



سیستم‌های عامل - پاییز ۱۳۹۹

گزارش کار

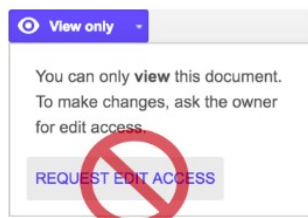
دانشکده مهندسی برق و کامپیوتر

نام و نام خانوادگی: علی بهاری

تاریخ: ۹۹/۹/۲۴

استاد:

دکتر مهدی کارگهی

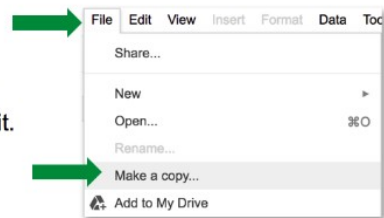


How to use this template:

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

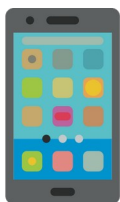
سوال چهارم

4

سوال پنجم

4

جدول دوم



مقدمه

در این تمرین به تحلیل داده‌هایی که از مشخصات و قیمت فروش گوشی‌های موبایل جمع‌آوری شده است پرداخته شده است. در ابتدا برنامه اقدام به خواندن و تجزیه مجموعه داده¹ی ارائه شده می‌کند و آنها را در حافظه خود ذخیره می‌کند. پس از استخراج داده‌ها و ویژگی‌های آنها، برنامه اقدام به نرمال‌سازی² داده‌ها و در نهایت اقدام به تعیین طبقه قیمتی گوشی‌ها می‌کند. این تمرین به دو روش این مسئله پیاده‌سازی شده است که در ادامه گزارش، نتایج حاصل آمده است.



- 1 Dataset
- 2 Data Normalization

پیاده‌سازی سری

سوال اول

چرا برای پیاده‌سازی یک برنامه بصورت چندریسه‌ای، بهتر است ابتدا این برنامه بصورت سری پیاده‌سازی شود؟

به دلیل این که بتوان قبل شروع به موازی سازی مشخص کرد کدام بخش های کد می تواند به صورت موازی پیاده سازی شود. در واقع با ساختار برنامه آشنایی بیشتری پیدا کنیم.

سوال دوم

با بررسی زمان اجرای بخش‌های مختلف برنامه، **Hotspot**³ های برنامه را مشخص کنید.
Extract کردن اطلاعات از فایل های **csv** – تبدیل داده ها به حالت **normalized** – به دست آوردن **accuracy** – به دست آوردن **ماکسیمم** و **مینیمم**

جدول اول

زمان‌های اجرای ۶ اجرای متوالی از برنامه و میانگین آن‌ها را بازای 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
0.019	0.015	0.022	0.013	0.036	0.016	0.014

3 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؟

شاید مشکل کار از نحوه تقسیم کار ها باشد پس باید بهتر این کار انجام شود. شاید مشکل از تعداد ریشه ها باشد یعنی ساخت ریشه ها که باعث ایجاد **overhead** می شود بهینه نباشد. شاید در قسمتی از برنامه بلاکی غیرمفید وجود دارد به این شکل که مثلا **join** ای داریم که قابل حذف شدن است یا حتی ممکن است مشکل از **lock** کردن ناحیه بحرانی باشد که به **busy working** برخورد کنیم و یا مدت زیادی در آن جا یک ریشه بلاک شود.

سوال چهارم

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

اشتراک داده های داخل برنامه بین ریشه ها - < متغیر ها به صورت **global** تعریف شدند تا همه ریشه ها به آن ها دسترسی داشته باشند.

هدر رفتن مقداری از تایم به خاطر تعداد **join** های زیاد - < بعضی از توابعی که در حالت سریال داشتیم با هم ادغام شدند تا در این صورت تعداد **join** ها کم تر و در نتیجه بلاک کوتاه تر شود.

به دست آوردن ماکسیمم و مینیمم کل ریشه ها - < در این حالت از **join** استفاده شده است تا بعد از این که هر ریشه ماکس و مین های هر ستون خود را به دست آورد بتوان ماکس و مین کلی را از بین این اعداد به دست آورد بعد از به دست آوردن این مقدار کلی دوباره ریشه ها به کار خود برای انجام کار های بعدی می پردازند که از ایت مقدار کلی در مراحل بعد بین ریشه ها استفاده خواهد شد و چون بین ریشه ها این مقدار کلی مشترک به صورت گلوبال در آمده است.

سوال پنجم

با توجه به تجربه‌ای که در پیاده‌سازی این تمرین بدست آوردید، به نظر شما در چه مواقعی از قفل⁴ در یک طراحی چندریسه‌ای ضروری است؟ تاثیر استفاده از قفل‌ها را بر روی کارایی⁵ سامانه بیان کنید.

وقتی که یک ریسه از داده‌های ریسه دیگر استفاده می‌کند که توسط آن ریسه‌ها قابل تغییر هستند.

وقتی که چند ریسه بخواهند یک داده را تغییر بدهند. در واقع ناحیه بحرانی باشد. اگر تعداد دفعات lock و unlock زیاد باشد و یا یک ریسه مدت زیادی پشت یک lock بماند کارایی پایین می‌آید.

جدول دوم

زمان‌های اجرای ۶ اجرای متوالی از برنامه و میانگین آن‌ها را بازای 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
0.010	0.008	0.013	0.010	0.009	0.011	0.010

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

میزان تسریع	میانگین زمان اجرای موازی	میانگین زمان اجرای سری
1.9	0.010	0.019

⁴ Lock

⁵ Performance