

# به نام خداوند جان و خرد

تمرین تحویلی سه  
درس رایانش ابری

امیرفاضل کوزه گر کالجی

9931099

## مسئله دایکسترا

در این مسئله از سه تابع بهره می‌بریم. Mapper, reducer, Dijkstra

تابع map، به ازای همسایه‌های نود ابتدایی، فاصله جدیدی را حساب کرده و آن را به همراه آن گره ذخیره می‌کند.

تابع reducer یک شی از فاصله‌هایی که پتانسیل کمترین فاصله بودن را دارند تهیه می‌کند.

و در نهایت تابع Dijkstra بررسی و مقایسه نهایی را انجام می‌دهد و مقادیر نتیجه را مشخص می‌کند.

## مسئله پیچ رنگ

تابع مپ در این مسئله، بر تمامی نودهای موجود در گراف پیمایش می‌کند. به ازای هر نود یک مقدار پیچ رنگی که باید contribute بکند را محاسبه کرده و آن را بر تعداد گره‌های همسایه تقسیم می‌کند. در نهایت لیستی از همکاری‌ها را برخواهد گرداند.

تابع reducer، با پیمایش بر لیست همکاری‌ها، مقدار پیچ رنگ‌های بعدی را حساب می‌کند.

# اسپارک

با اجرای run\_cluster.sh کانتینر های خواسته شده را می‌سازیم.

```
What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview
[+] Building 0.0s (0/0)
[+] Running 14/14
✓ Volume "docker_hadoop_datanode1"      Created
✓ Volume "docker_hadoop_datanode2"      Created
✓ Volume "docker_hadoop_namenode"       Created
✓ Volume "docker_hadoop_historyserver"   Created
✓ Container spark-master                 Started
✓ Container namenode                     Started
✓ Container jupyter-notebook             Started
✓ Container spark-worker2                Started
✓ Container spark-worker1                Started
✓ Container datanode1                    Started
✓ Container datanode2                    Started
✓ Container resourcemanager               Started
✓ Container historyserver                 Started
✓ Container nodemanager1                  Started
```

در ادامه به توضیح کانتینر ها به صورت مختصر خواهیم پرداخت:

- **Resource manager**  
این کامپوننت، وظیفه مدیریت منابع، تخصیص دادن آنها و نظار بر cpu سیستم را بر عهده دارد.
- **History server**  
این کامپوننت، وظیفه ردیابی جاب های پایان یافته در محیط هدوپ را برعهده دارد. و در مورد آنها جزئیاتی را درمورد تاریخچه شان نگهداری می‌کند.
- **Data node**  
این کامپوننت به طور کلی، مسئول نگهداری داده های هدوپ در سیستم های توزیع شده می‌باشد.
- **Nodemanager**  
این کامپوننت بخشی از YARN می‌باشد. این کامپوننت بر هر ماشین کلاستر اجرا می شود و وظیفه مدیریت کانتینر ها در آن ماشین را بر عهده دارد.

- Namenode

این کامپوننت همانند مرکز نگهداری متادیتا در هدوپ می باشد. داده هایی مانند name space و control access ها توسط این کامپوننت نگهداری می شود.

Notebook questions:

- DAG scheduler

یکی از مولفه های مهم اپاچی اسپارک می باشد. مخفف directed acyclic graph می باشد. هدف مهم آن، تبدیل روند های فرایندی داده کاربر به مراحل است که بتوان به صورت توزیع شده آن را اجرا کرد.

وقتی یک جاب در اسپارک ثبت می شود، این مولفه آن را به تسک های کوچک و قابل مدیریت تری تقسیم می کند.

DAG به گرافی از RDD ها مبدل می شود و عملیات روی آنها اعمال می شود.

پس از اینکه مراحل یا stage ها تعریف شدند، DAG scheduler عملگر های درون سیستم را به یکدیگر به مانند یک پایپلاین وصل می کند و یک پلن اجرایی را آماده می کند. سپس تمامی این تسک ها را به task scheduler می سپارد.

- نحوه عملکرد شافلینگ

در آپاچی اسپارک، عمل shuffling به عنوان روشی برای توزیع مجدد داده ها استفاده می شود. این توزیع مجدد به گونه ای است که داده ها را در گره های مختلف توزیع می کند.