به نام پروردگار هدایت کننده به راه راست

دانشگاه اصفهان
ساختمان داده – دکتر رمضانی
پاییز ۰۵-۰۴

پروژه نهایی - سیستم بلاک‌چین

طراحان پروژه : امیرعلی گلی - صادق جباری

مبحث : درخت‌ها و گراف‌ها

اهداف پروژه :

- کار با ساختمان داده درخت (مانند درخت‌های متعادل باینری جستجو)
- پیاده‌سازی عملیات‌های پرکاربرد درخت‌ها و گراف‌ها (مانند مرتب‌سازی توپولوژیکی در DAG)
- آشنایی با مفاهیم بلاک‌چین و سیستم‌های توزیع‌شده

در این پروژه قرار است با استفاده از ساختمان داده‌های پیشرفته مانند درخت‌های متعادل باینری جستجو (Balanced Binary Search Trees) و گراف غیرمدور جهت‌دار (DAG) برای ممپول، یک هسته نرم‌افزاری ساده برای یک ماینر بیت‌کوین پیاده‌سازی کنید. این سیستم شامل مدیریت تراکنش‌ها، ممپول، استخراج بلاک است.

توضیحات کامل پیاده‌سازی گراف DAG و کاربردهای آن در بلاک‌چین را می‌توان از لینک‌های زیر مطالعه نمود:

- https://www.geeksforgeeks.org/detect-cycle-in-a-graph/
- https://en.wikipedia.org/wiki/Directed_acyclic_graph

همچنین برای آشنایی با بلاک‌چین و ممپول می‌توانید از لینک‌های زیر استفاده کنید:

- https://bitcoin.org/en/developer-guide#mempool
- https://en.wikipedia.org/wiki/Blockchain

۱- مقدمه

شما در ابتدا باید ساختمان داده‌های پایه مانند درخت‌های متعادل باینری جستجو برای امتیازدهی اجداد و نوادگان در ممپول، و گراف DAG برای مدیریت وابستگی‌های تراکنش‌ها پیاده‌سازی کنید. سپس با استفاده از این ساختمان داده‌ها، یک نود بلاک‌چین CLI (Command-Line Interface) بسازید که عملیات‌هایی مانند افزودن تراکنش به ممپول، استخراج بلاک را پشتیبانی کند.

این پروژه به عنوان یک هسته نرم‌افزاری ماینر بیت‌کوین طراحی شده است، جایی که ممپول به صورت DAG ذخیره می‌شود و تراکنش‌ها بر اساس نرخ کارمزد (fee rate) اولویت‌بندی می‌شوند.

برای مثال، در تست‌کیس۱ (Parent-Child Dependency)، وابستگی بین تراکنش‌ها بررسی می‌شود که چگونه یک تراکنش فرزند می‌تواند به تراکنش والد وابسته باشد، و در بلاک استخراج‌شده، ترتیب توپولوژیکی رعایت می‌شود.

۲- ویژگی‌ها

۲-۱- مدیریت تراکنش‌ها

- محاسبه کارمزد تراکنش بر اساس ورودی‌ها و خروجی‌ها.

۲-۲- مدیریت ممپول (Mempool)

- پیاده‌سازی ممپول به صورت DAG برای ذخیره تراکنش‌های تاییدنشده با وابستگی‌ها (مانند سناریو Multi-Level Chain در۲).
- امتیازدهی اجداد و نوادگان بر اساس نرخ کارمزد (ancestor/descendant fee rate) با استفاده از دو درخت متعادل باینری جستجو.
- افزودن تراکنش به ممپول (AddTransactionToMempool) و خارج کردن تراکنش‌ها (EvictMempool) بر اساس اولویت descendant fee rate (مانند تست ۴ که تراکنش standalone با کارمزد پایین حذف می‌شود).

۲-۳- استخراج بلاک

- تنظیم سختی (SetDifficulty) و استخراج بلاک (MineBlock) با الگوریتم Proof-of-Work (PoW) و جستجوی nonce تا رسیدن به هش معتبر.

- انتخاب حریصانه تراکنش‌ها از ممپول بر اساس ancestor fee rate، با رعایت حد اندازه بلاک (۱۰۰۰۰ بایت) و ترتیب توپولوژیکی (مانند سناریو CPFP در تست۳ و Diamond-Shaped DAG در تست۶).

۲-۵- پشتیبانی از DAG پیشرفته

- بهبود مدیریت وابستگی‌ها در DAG برای سناریوهای پیچیده مانند Multiple Children (تست۵).

۳- ساختار پروژه

پروژه به چندین بخش اصلی تقسیم می‌شود:

۱- بخش Core Data Structures:

- پیاده‌سازی درخت‌های متعادل باینری جستجو برای امتیازدهی ancestor/descendant در ممپول.
- سریال‌سازی داده‌ها با تابع درهم‌سازی سفارشی (توضیحات آن در ویدئوها ذکر شده است)

۲- بخش Mempool DAG:

- پیاده‌سازی گراف DAG برای تراکنش‌ها با عملیات افزودن، حذف و مرتب‌سازی توپولوژیکی.
- مدیریت وابستگی‌ها و امتیازدهی بر اساس کارمزد
پشتیبانی از عملیات EvictMempool برای حذف کم‌ارزش‌ترین تراکنش‌ها و نوادگان‌شان.

۳- بخش Blockchain and Mining:

- استخراج بلاک با انتخاب حریصانه تراکنش‌ها و PoW
- CLI برای دستورات مانند MineBlock

ویژگی‌های امتیازی:

- تحقیق و بررسی ساختمان‌داده Merkle Tree و کاربرد آن در بلاک‌ها و پیاده‌سازی آن (اضافه کردن فیلد Merkle Root به Header بلاک)

نکات تکمیلی :

- این پروژه به صورت گروه‌های تا دو نفره باید پیاده‌سازی شود.
- بستر پیاده‌سازی پروژه روی گیت‌هاب می‌باشد.
- سعی کنید هر یک از بخش‌ها را در یک کامیت جداگانه انجام دهید.
- رعایت اصول کدنویسی تمیز بخش بسیار زیادی از نمره را به خود اختصاص می‌دهد و درصورتی که کد کاملا به شکل غیراصولی پیاده‌سازی شده باشد، تحویل گرفته نمی‌شود.
- استفاده از هر زبان، فریمورک و رابط‌های گرافیکی کاملا آزاد است.
- تست‌کیس‌ها (فایل‌های ۱ تا ۶) باید به طور کامل پاس شوند و خروجی‌ها با نمونه‌های ارائه‌شده مطابقت داشته باشند.
- تصاویر توضیحی پروژه (مانند ساختار بلاک یا فرآیند ماینینگ) از فایل blockC.pdf استخراج شده و می‌توان در مستندات گیت‌هاب استفاده کرد.