

تمرین دوم - الگوریتم های حریصانه

در این تمرین، قصد داریم که به سری الگوریتم های حریصانه (greedy) در کنار یک کاربرد جالب از آن بپردازیم. در این تمرین شما قرار است که الگوریتمی را پیاده سازی کنید که بتواند به سیستم زنجیره بلاکی (blockchain) ما کمک کند تا اطلاعات رد و بدل شده در مجموع زمان کمتری را به خود اختصاص دهند.

شرح تمرین:

در این تمرین ما با یک نمونه ساده از یک سیستم زنجیره بلاکی آشنا خواهیم شد که در آن در ابتدا تعداد گره (node) های سیستم به همراه هزینه فرضی ارتباط بین آن ها، مشخص شده و هر یک از آنها در سیستم به عنوان یک شی از کلاس Node خواهند بود. سپس به صورت متوالی، به اندازه یک مقدار زمان از پیش تعیین شده، تراکنشی در سیستم رخ خواهد داد که در آن مقداری سکه از یک sender به یک receiver خواهد رفت. پس از رخ دادن تراکنش، گره هایی که این تراکنش را انجام داده اند، ابتدا می بایست این تراکنش را به استخر حافظه خود اضافه کرده و سپس به باقی گره های موجود در سیستم این تراکنش را اطلاع رسانی کنند تا آنها نیز این تراکنش را در استخر حافظه خود قرار دهند.

عمل استخراج همانند رخ دادن یک تراکنش، به صورت تصادفی و بعد از گذشت زمانی رخ خواهد داد. در این اتفاق، یک گره به صورت تصادفی انتخاب شده که گره خوش شانس ما می باشد. این گره در ابتدا، استخر حافظه خود را با استفاده از کلید خصوصی خود، امضای دیجیتال کرده و آن را به عنوان یک بلاک، به دیگر گره های موجود در سیستم اطلاع رسانی می کند.

در این سیستم، بعد از گذشت زمانی از پیش تعیین شده، در ابتدا تعدادی گره به صورت تصادفی به سیستم اضافه شده و به هر یک از گره های سیستم با فراخوانی تابع `update_costs` هر گره، اطلاع رسانی انجام خواهد شد تا عمل آپدیت کردن مقادیر فاصله های نود های موجود را انجام دهند.

کاری که شما لازم در این تمرین انجام دهید این است که الگوریتمی بنویسید که مقادیر `costs` را جوری تنظیم کند که عمل `gossiping` در زمان استخراج بهینه شود. برای سنجش این موضوع توابع آماده ای در اختیار شما قرار خواهد گرفت. شما تنها می بایست که الگوریتم برورسانی مقادیر هزینه های گره ها را پیاده سازی کنید.

هر یک از گره های سیستم بلاکچین، تعدادی مشخصه دارند که در زیر لیست شده اند:

- `blocks`: هر یک از گره های سیستم ها، لیستی از بلاک ها خواهند داشت که بلاک های استخراج شده را در خود ذخیره خواهند کرد.
- `mem_pool`: استخر حافظه یا `memory pool` لیستی است که هر گره، هریک از تراکنش های رخ داده در سیستم را در خود نگه داری خواهد کرد. این استخر حافظه تا رخ دادن استخراج بعدی، تراکنش های رخ داده در سیستم را در خود ذخیره خواهد کرد.
- `costs`: این مشخصه، لیستی است که مقدار فاصله تا هر یک از گره های موجود در سیستم را در خود ذخیره می کند. اگر گره ای با گره دیگری متصل نباشد، در ابتدا این مقدار برابر ∞ خواهد بود.

نکات که می‌بایست برای شروع و ارسال تمرین آن‌ها را رعایت فرمایید:

۱- شما هیچ نیازی به پیاده سازی سیستم زنجیره بلاکی ندارید و کد های اولیه آن در اختیار شما قرار خواهد گرفت و شما تنها می‌بایست الگوریتم بروزرسانی مقادیر costs را پیاده سازی نمایید.

۲- برای تحویل این تمرین می‌بایست یک سند به فرمت pdf به همراه کد ها، تحویل داده شود که در آن، جزئیات پیاده سازی و الگوریتم مورد نظر به همراه دلیل انتخاب این الگوریتم، و رویکرد حل مسئله شما مفصل توضیح داده شده باشد در غیر اینصورت مقداری از نمره این تمرین کسر خواهد شد.

توجه: کد های اولیه این تمرین به زودی در اختیار دانشجویان قرار خواهد گرفت.

* به منظور پاسخ به سوالات و رفع اشکالات دانشجویان، متقابلاً زمانی برای برگزاری کلاس حل تمرین اعلام خواهد شد.

موفق باشید.