



سیستم‌های توزیع شده

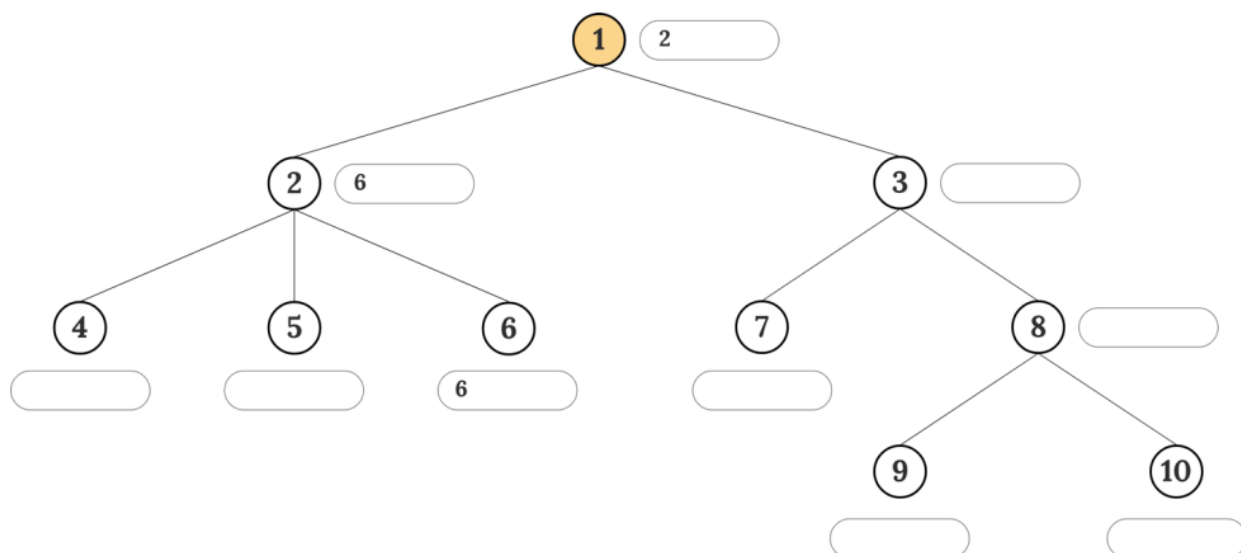
تمرین سری پنجم

دکتر کمندی

پارسا محمدپور

پرسش یک

شکل زیر یک حلقه ریموند را نشان می‌دهد. گره یک توکن را در اختیار دارد. اعداد کنار هر گره نشان‌گر صف کنونی هستند.



الف) فرض کنید در حالیکه گره یک توکن را همچنان در اختیار دارد، گره‌های نه، هفت و پنج به ترتیب درخواست ورود به ناحیه بحرانی بدهند. ورودی‌های صف‌های تمام گره‌ها را پس از پردازش درخواست این گره‌ها بدست آورید.

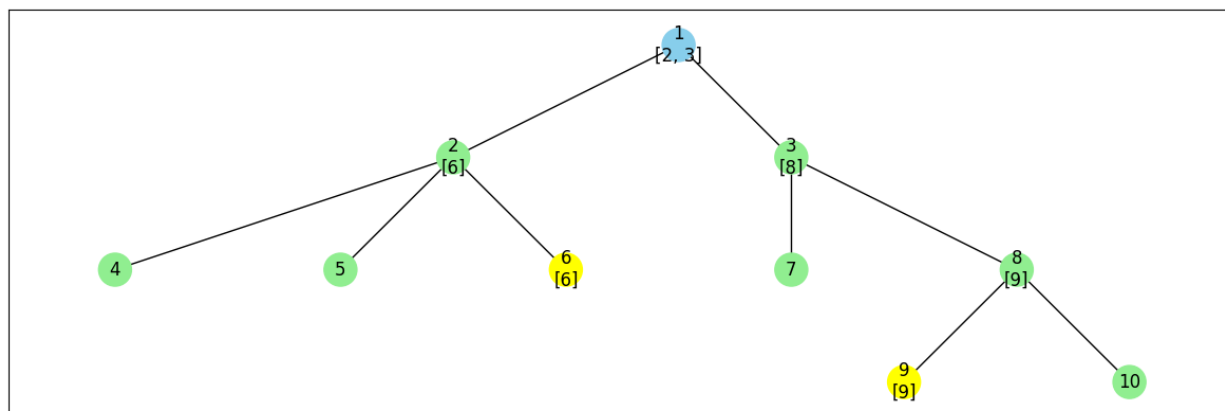
ابتدا یک توضیح کلی در رابطه با عملکرد الگوریتم ارائه می‌دهیم. این الگوریتم، یکی از انواع الگوریتم‌های مبتنی بر توکن^۱ می‌باشد. در این الگوریتم، ما ابتدا باید یک **درخت** (دلیل استفاده از درخت اسن است که نباید در این الگوریتم دور وجود داشته باشد؛ چون در صورت وجود داشتن دور، امکان رخداد بن‌بست^۲ بوجود می‌آید) از اتصال سایت‌ها داشته باشیم. تنها سایتی (گره‌ای) که توکن را در اختیار داشته باشد توانایی ورود به ناحیه بحرانی را دارد. هر سایت، پس از تمام شدن کارش با ناحیه بحرانی و پس از خروج از آن، توکن را به سایتی که در ابتدای صفش قرار دارد پاس می‌دهد و ساختار درختی که در این الگوریتم وجود دارد، عوض می‌شود؛ به طوریکه سایتی که قبل از آن توکن را در اختیار داشت، الآن، فرزند سایتی می‌شود که در حال حاضر گره را در اختیار دارد. حال با توجه به اسن توضیحات و مفروضات داده شده، به سراغ حل این مسئله می‌رویم. (برای این مسئله یک کد هم زده شده است که می‌توان آن را در این [لینک](#) مشاهده کرد) حالا در این الگوریتم برای یک حالت خاص، می‌توانیم دو رویکرد مختلف را در پیش بگیریم.

در این وضعیت، سایت نه ابتدا درخواست خود را می‌دهد. پس باید در صف مربوط به خود سایت نه، مقدار نه وارد شود. سپس سایت هشت نیز درخواست را از سایت نه دریافت کرده و مقدار نه را در صف خودش جای می‌دهد و سپس درخواست را به سایت سه می‌دهد. سایت سه نیز پس از دریافت این درخواست، سایت هشت را در صف خود قرار می‌دهد و این درخواست را برای سایت یک

^۱ Token-based algorithm

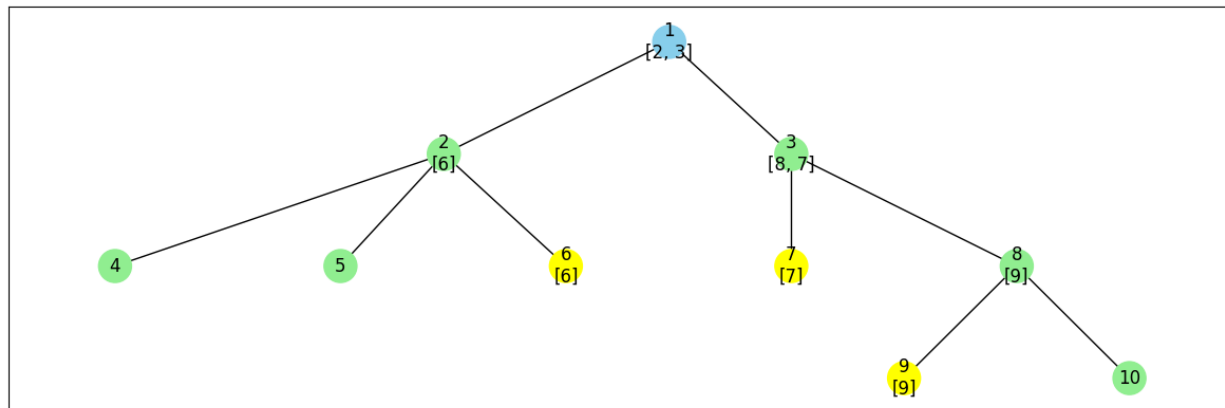
^۲ Deadlock

ارسال می‌کند. سایت یک نیز پس از دریافت این درخواست، سایت سه را در صف خود اضافه می‌کند. سپس شکل سایت‌ها به همراه صف هایشان به صورت زیر می‌شود:



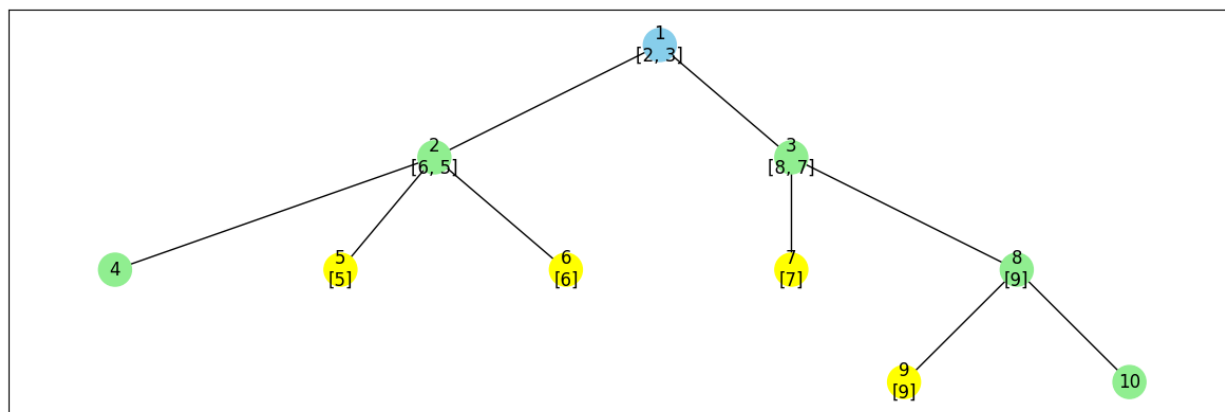
۱) این شکل، گراف سایت‌ها و صف‌هایشان پس از دریافت درخواست سایت نه به وسیله همه سایت‌ها است. در این شکل سایتی که توکن را در اختیار دارد، با رنگ آبی، سایتی که درخواست توکن را دارد، با رنگ زرد و بقیه سایت‌ها هم با رنگ سبز نشان داده شده‌اند.

سپس طبق صورت سوال، سایت هفت درخواست خود را برای سایت سه ارسال می‌کند و سایت سه سایت هفت را در صف خود قرار می‌دهد. سپس سایت سه نیز درخواست خود را برای سایت یک ارسال می‌کند و سایت یک نیز آن را دوباره در صف خود قرار می‌دهد. پس شکل گراف حاصل و صف‌های سایت‌ها به صورت زیر می‌شود:



۲) شکل سایت‌ها و صف‌هایشان پس از دریافت درخواست سایت هفت

سپس سایت پنج درخواست خود را به سایت دو ارسال می‌کند و سایت دو علاوه بر اضافه کردن سایت پنج، درخواست سایت پنج را به سایت یک نیز ارسال می‌کند. سپس سایت یک هم سایت دو را در صف خود اضافه می‌کند. پس شکل سایت‌ها و صف‌هایشان به صورت زیر می‌شود:

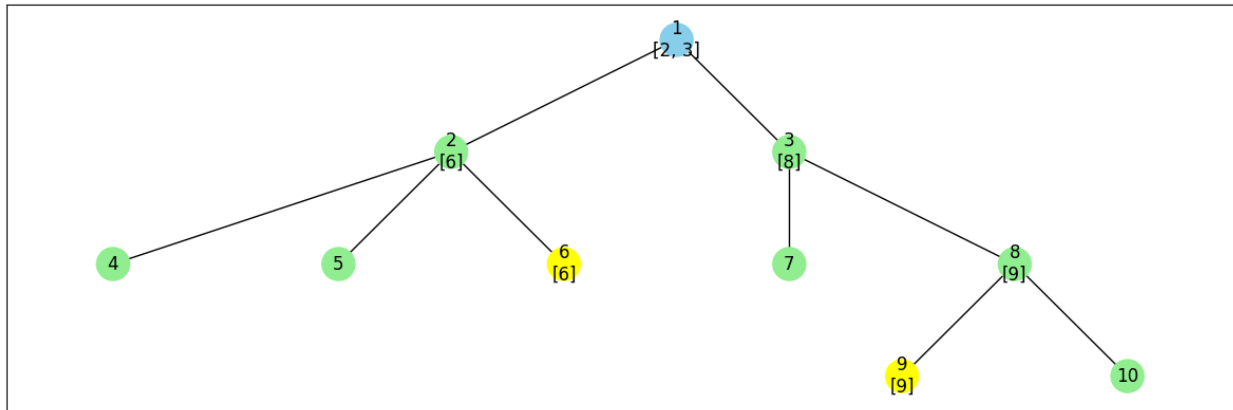


۳ شکل سایت‌ها و صف‌هایشان پس از دریافت درخواست سایت پنج

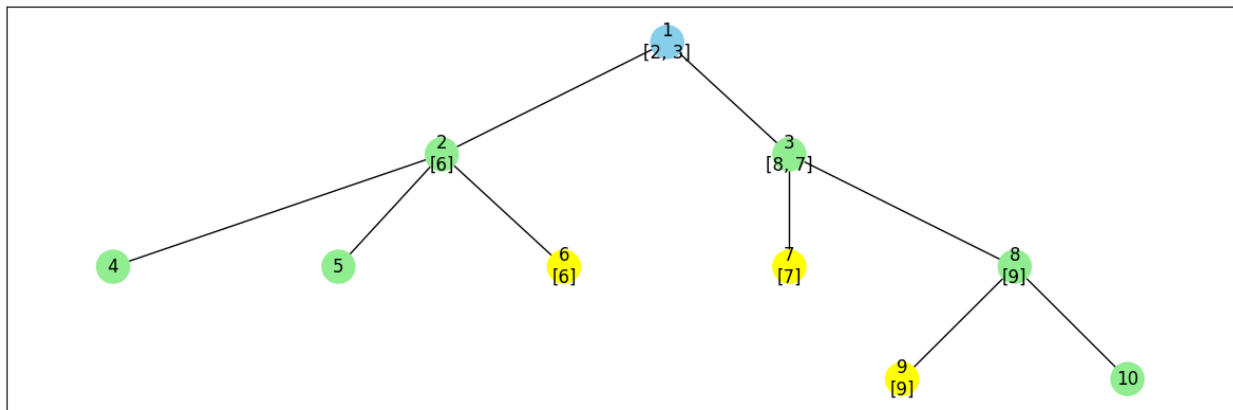
حالا در این قسمت با این چالش روبرو نشدیم، ولی حالا الآن وقتی که برای قسمت بعدی سایت یک می‌خواهد توکن را به سایت دو بدهد، چون در صف خودش چند سایت وجود دارد باید چه کند؟ دو رویکرد را می‌توانیم مدنظر قرار بدهیم. در یکی از این رویکردها، همیشه درخواست زودتر، زودتر به ناحیه بحرانی می‌رسد، در حالت دیگر درخواستی که نزدیک‌تر (نه دقیقا نزدیک‌تر، یعنی در همان ناحیه باشد و نیازی نداشته باشد تا برای اینکه توکن به آن برسد از سایت بالایی رد شود) است، سریع‌تر به ناحیه بحرانی وارد می‌شود. برای مثال وقتی در این حالت، سایت یک می‌خواهد توکن را به سایت دو بدهد، این دو حالت را داریم: می‌توانیم برای سادگی سایت یک را در انتهای صف مربوط به سایت دو قرار دهیم و ادامه بدهیم (یعنی اگر سایتی که در حال واگذاری توکن است، صفش خالی نیست، خودش را در انتهای صف دیگری قرار دهد)؛ یا راه دوم این است که با توجه به زمان منطقی درخواست سایت‌ها، آن را در جایی که سریع‌ترین درخواستش آمده است قرار دهیم. یعنی برای مثال اگر اولین درخواست موجود در سایت یک هنگام تحویل توکن به سایت دو، برای سایت سه و باید زودتر از درخواست سایت دو (آن درخواستی که برای سایت پنج است، نه آن درخواستی که برای سایت شش است) انجام شود، در این حالت، ما می‌توانیم تضمین کنیم که همیشه درخواستی که زودتر آمده است، زودتر هم پردازش شود و توکن را در دست بگیرد.

هر دوی این حالات در کد آورده شده پیاده‌سازی شده‌اند. در این قسمت، نتایج حاصل از این دو تا رویکرد، تفاوتی نمی‌کرد و عکس‌هایی که تا اینجا قرار داده شده‌اند، برای حالتی است که صرفا به ته صف اضافه می‌کردیم. (یعنی حالت اول) ولی در این قسمت و قسمت‌های بعد، عکس‌های حاصل از پیاده‌سازی رویکرد دوم، که مبتنی بر زمان ارسال درخواست است، نیز آورده شده‌اند؛ اما دیگر توضیحی برایشان ارائه نمی‌شود.

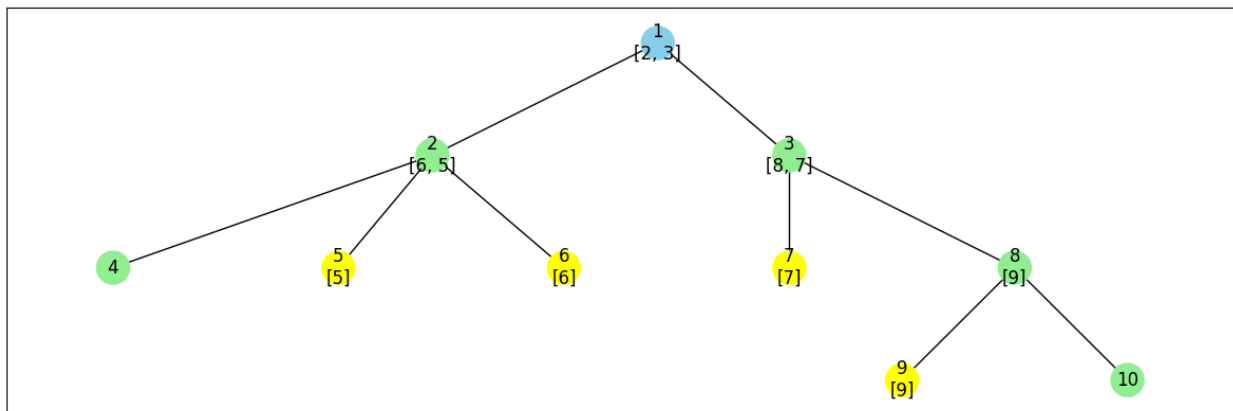
عکس‌های حاصل از پیاده‌سازی روش مبتنی بر زمان منطقی:



۴ وضعیت سایت‌ها پس از پردازش درخواست سایت نه



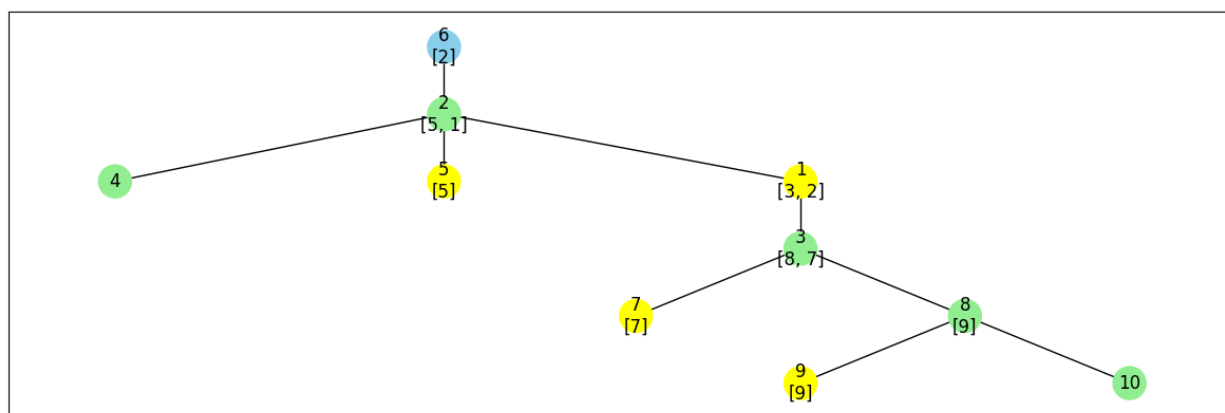
۵ وضعیت نهایی سایت‌ها پس از پردازش درخواست سایت ده



۶ وضعیت نهایی سایت‌ها پس از پردازش درخواست سایت پنج

ب) فرض کنید درخواست دیگری برای ورود به ناحیه بحرانی از طرف گره‌ای ارسال نشود. ورودی‌های صف‌های گره‌های مرتبط را وقتی توکن به گره شماره شش می‌رسد، به دست آورید.

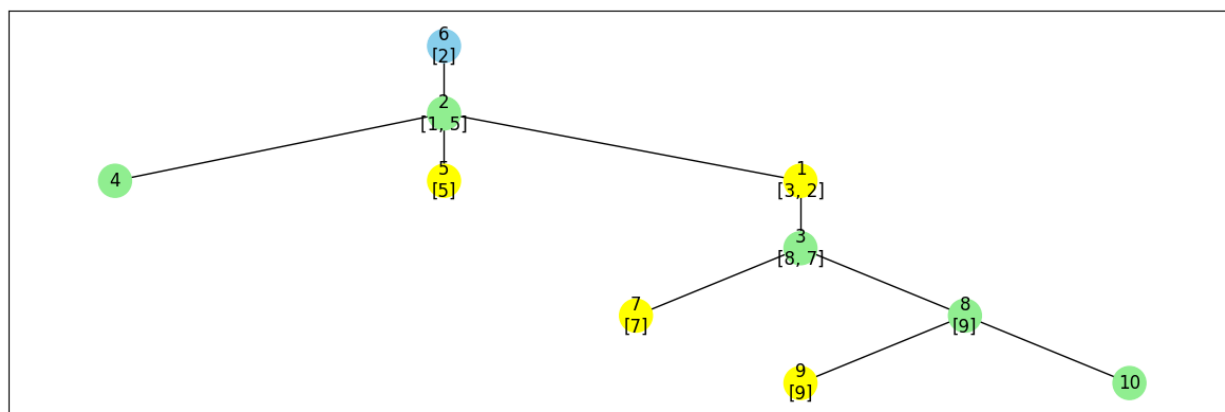
حالا با توجه به اینکه کار سایت یک با ناحیه بحرانی تمام شده‌است، پس از لیست خود اولین سایت را برمیدارد و توکن را به آن می‌فرستد و اما با توجه به اینکه در حال حاضر، صف آن، شامل تعدادی سایت هست همچنان، سایت دو مقدار سایت یک را نیز در انتهای صف خودش قرار می‌دهد. (حالت اول) سپس سایت دو نیز پس از دریافت توکن، با توجه به اینکه در ابتدای صفش، سایت شش قرار دارد، توکن را برای سایت شش ارسال می‌کند و با توجه به اینکه صف خودش خالی نیست، خودش (سایت دو) هم در انتهای صف مربوط به سایت شش، قرار می‌گیرد. پس در انتها، توکن در سایت شش قرار می‌گیرد و در صف سایت دو، سایت یک قرار دارد و در صف سایت شش هم، سایت دو قرار دارد. شکل حاصل به صورت زیر می‌شود:



۷ شکل سایت‌ها و درخواست کنندگان که منتظر دریافت توکن هستند و سایت شش که رادای توکن است

پس بدین ترتیب، سایت شش توکن را دریافت می‌کند و در صف آن، سایت دو قرار دارد و در صف سایت دو، سایت‌های پنج و یک قرار دارند و به همین ترتیب سایت‌های موجود در صف سایت‌های دیگر هم مشخص شده‌اند.

حالا مطابق با توضیحات ارائه شده در قسمت الف، نتیجه حاصل از پردازش درخواست‌ها مبتنی بر زمان ارسال درخواست‌شان را هم در ادامه قرار می‌دهیم.

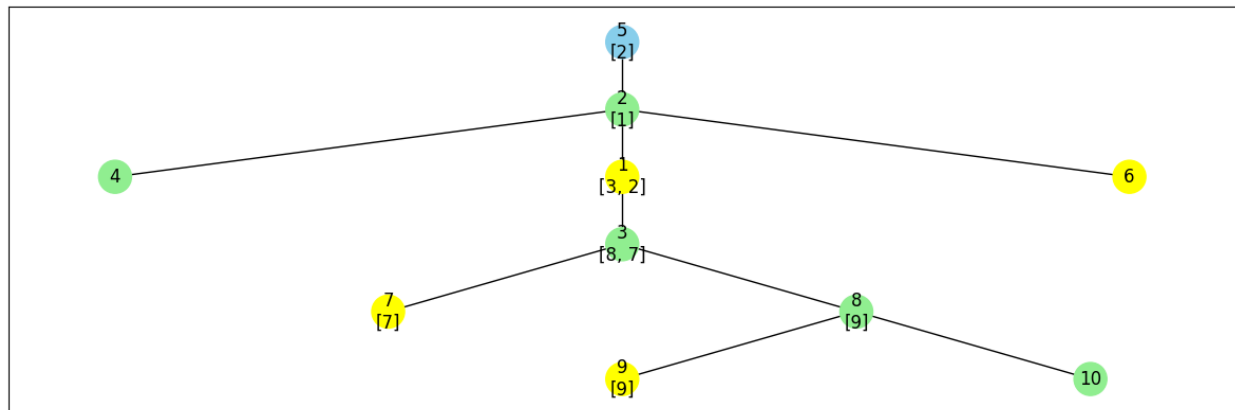


۸ وضعیت نهایی سای‌ها پس از اتمام کار سایت یک و ارسال توکن برای سایت شش

ج) قسمت قبل را برای وقتی که توکن به گره نه می‌رسد، تکرار کنید.

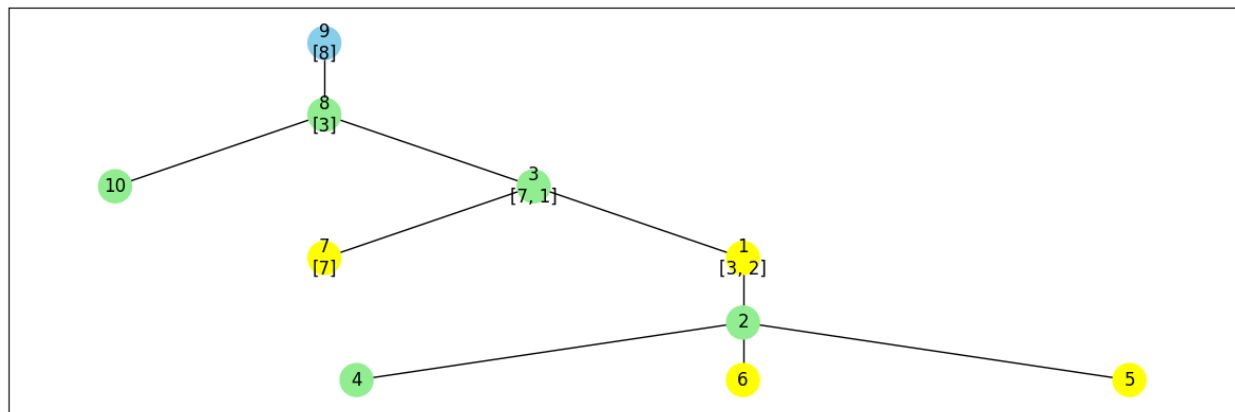
همین کار را به قدری انجام می‌دهیم تا توکن به گره شماره نه برسد. پس به ترتیب گفته‌ده جلو می‌رویم و مطابق الگوریتم مراحل را طی می‌کنیم.

در اولین مرحله، پس از اتمام کار سایت شش با ناحیه بحرانی، سایت شش توکن را انتقال می‌دهد تا به سایت پنج برسد. (چون ما صرفاً بدون در نظر گرفتن زمان منطقی ارسال هر درخواست، سایتی که توکن را می‌دهد به ته صف سایت گیرنده اضافه می‌کنیم، سایت پنج، سایت بعدیست. درحالی‌که اگر با زمان منطقی جلو می‌رفتیم، سایت بعدی باید سایت نه می‌بود) سایت پنج توکن را در اختیار دارد. شکل حاصل پس از انجام این مرحله و قرارگیری توکن در سایت پنج، به صورت زیر می‌شود:



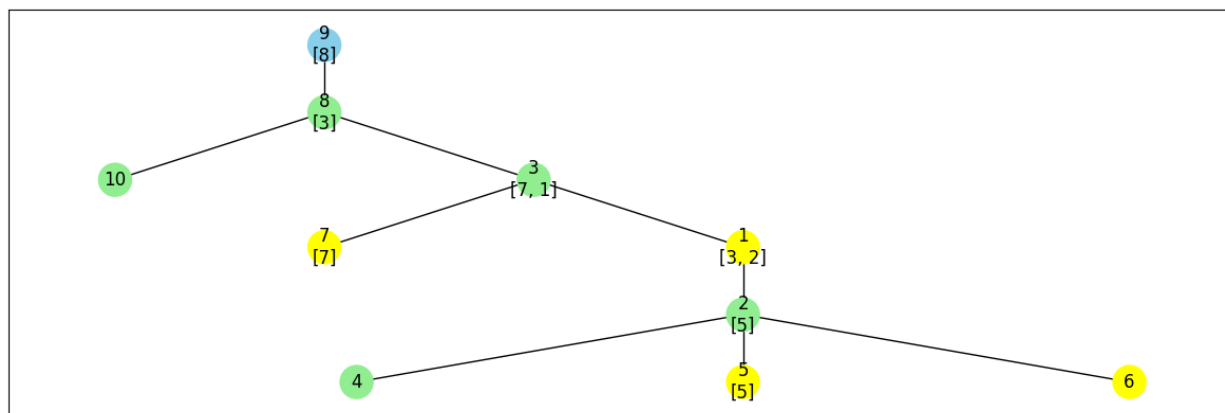
۹ شکل سایت‌ها و صف‌هایشان. در این لحظه سایت پنج توکن را در اختیار دارد و می‌تواند وارد ناحیه بحرانی شود

سپس یک مرحله دیگر جلو می‌رویم. این سری توکن به سایت نه می‌رسد. چون اولویت با اولین سایتی است که در صف هر سایت قرار دارد هنگامی که آن سایت توکن را در اختیار بگیرد. شکل حاصل به صورت زیر می‌شود:



۱۰ شکل نهایی سایت‌ها و صف‌هایشان پس از در اختیار گرفتن توکن به وسیله سایت نه

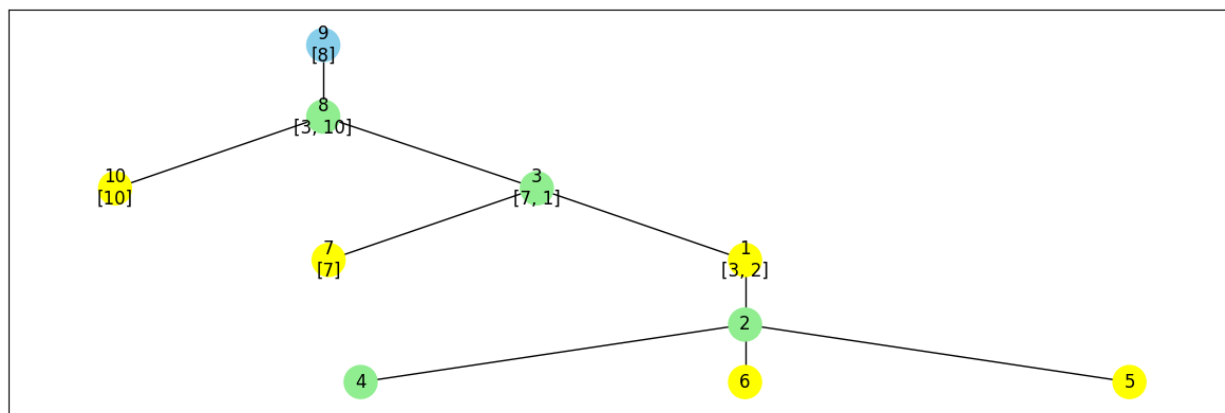
در این قسمت نیز مشابه با قسمت‌های پیشین، نتیجه حاصل از اجرای الگوریتم مبتنی بر زمان را هم قرار می‌دهیم. اما در این قسمت برخلاف قسمت‌های قبلی، نتیجه تغییر می‌کند و درخواست سایت نه، زودتر اجرا می‌شود.



! نتیجه الگوریتم مبتنی بر زمان پس از اتمام پردازش سایت شش و ارسال آن به سایت نه

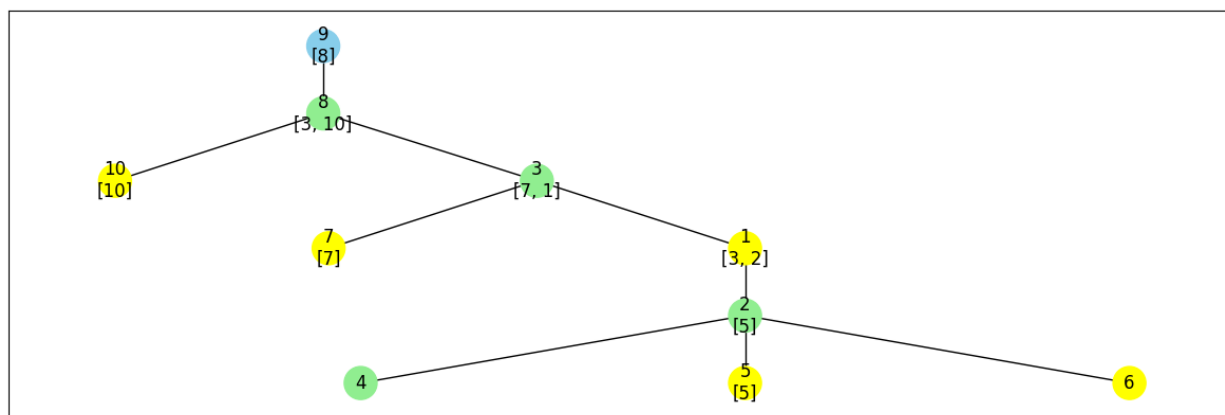
د) فرض کنید وقتی توکن در اختیار گره نه است، گره ده درخواست ورود به ناحیه بحرانی می‌دهد. ورودی‌های صف گره‌های مرتبط را بدست آورید.

در این حالت، سایت نه دارای توکن است و در صف خود سایت هشت را دارد. (سایت موجود در صف هر سایت و اطلاعات دیگر در شکل ۱۰ موجود است). سپس سایت ده می‌خواهد وارد ناحیه بحرانی شود. برای این مورد درخواست خود را به سایت هشت ارسال می‌کند. (سایت هشت در حال حاضر در درخت این الگوریتم، پدر این سایت می‌باشد) سپس سایت هشت هم درخواست را برای سایت نه می‌فرستد. شکل نهایی به صورت زیر می‌شود:



۱۲ شکل نهایی سایت‌ها پس از اینکه سایت ده درخواست ورود خود را ارسال کرد

در این قسمت نیز مشابه با قسمت‌های پیشین، نتیجه حاصل از اجرای الگوریتم مبتنی بر زمان را هم قرار می‌دهیم. اما در این قسمت برخلاف قسمت‌های قبلی، نتیجه تغییر می‌کند و درخواست سایت نه، زودتر اجرا می‌شود.



۱۳ نتیجه ارسال درخواست توسط سایت ده برای سایت نه که دارنده توکن است

پرسش دو

گروهی از سیستم‌های توزیع شده P_1, P_2, P_3 و P_4 را در نظر بگیرید که یک شئی را به شاتراک می‌گذارند و برای مدیریت انحصار متقابل^۱، از الگوریتم ریکارتی-آگراوالا^۲ استفاده می‌کنند. P_1 در حال حاضر در ناحیه بحرانی است و هیچ گره دیگری در حالت "درخواست"^۳ نیست. اکنون درخواست‌های P_4, P_2 و P_3 (به ترتیب) برای ورود به همان ناحیه بحرانی را در نظر بگیرید.

الف) وضعیت "''"، "'' و ... موجود در صف هر سیستم را مشخص کنید.

ب) اکنون P_1 از ناحیه بحرانی خارج شده و به تمام گره‌های مربوطه اطلاع می‌دهد که ناحیه بحرانی آزاد شده است. در این مرحله، وضعیت ورودی‌های صف را در هر سیستم نشان دهید.

¹ Mutual Exclusion

² Ricarti-Agrawala

³ Wanted

پرسش سه

در خصوص احتمال رخداد بن‌بست در الگوریتم میکاوا^۱ بحث کنید.

^۱ Mackawa