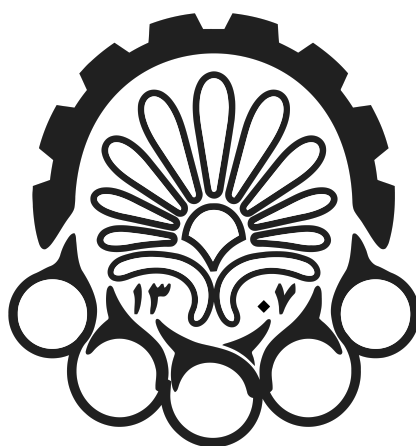


معماری افزاره‌های شبکه دکتر صبائی



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

رضا آدینه پور ۴۰۲۱۳۱۰۵۵

تمرین سری دوم

۲۰ آبان ۱۴۰۳

سوال اول

فرض کنید یک مسیریاب با استفاده از الگوریتم تطابق بیشترین طول پیشوند (Longest Prefix Matching)، جلورانی (Forwarding) بسته‌ها را انجام می‌دهد. در صورتی که جدول جلورانی مسیریاب به صورت زیر باشد:

Prefix	Next Hop
1010*	A
101*	B
101011*	C
100*	D

گام بعدی بسته‌های دریافتی با آدرس‌های مقصد زیر را بدست آورید:

• 10101101

پاسخ

طولانی ترین تطابق با 101011* است بنابراین مقصد C است.

• 10111101

پاسخ

طولانی ترین تطابق با 101* است بنابراین مقصد B است.

• 10001101

پاسخ

طولانی ترین تطابق با 100* است بنابراین مقصد D است.

سوال دوم

روش‌های مختلفی برای جستجوی آدرس IP در جدول جلورانی وجود دارد. هر کدام از این روش‌ها ویژگی‌هایی دارند.

- اهداف اصلی یک روش جستجوی آدرس IP چیست؟

پاسخ

اهداف اصلی یک روش جستجوی آدرس IP شامل موارد زیر است:

- تطابق سریع و دقیق: یافتن سریع و دقیق‌ترین تطابق (Longest Prefix Match) برای آدرس IP، به طوری که بسته به درستی و بدون تأخیر به مقصد هدایت شود.
- بهره‌وری بالا: استفاده بهینه از منابع سخت‌افزاری مانند حافظه و پردازشگر، تا روش جستجو بتواند با کمترین هزینه پردازشی و مصرف حافظه کار کند.
- قابلیت مقیاس‌پذیری: روش باید بتواند با افزایش اندازه جدول جلورانی و تعداد آدرس‌ها، همچنان به خوبی کار کند و عملکرد مطلوبی داشته باشد.
- پایداری و مقاومت در برابر تغییرات: امکان تطبیق و بروزرسانی آسان جدول جلورانی با ورود و خروج آدرس‌ها و یا تغییر مسیرها.

- معیارهای ارزیابی یک روش جستجوی آدرس IP کدام‌اند و چگونه می‌توان کارایی یک روش جستجوی آدرس IP را در شبکه‌های بزرگ ارزیابی کرد؟ (پاسخ خود را با توجه به مواردی مانند کارایی حافظه، زمان جستجو، مقیاس‌پذیری و موارد مشابه دیگر توضیح دهید.)

پاسخ

- کارایی حافظه:

- * میزان استفاده از حافظه یکی از معیارهای مهم برای ارزیابی روش‌های جستجو است. روش‌های کارا از حافظه بهینه استفاده می‌کنند، به‌ویژه در جدول‌های جلورانی بزرگ.
- * روش ارزیابی: اندازه جدول و ساختار داده‌های ذخیره شده برای هر روش را مقایسه می‌کنیم. روش‌هایی که از ساختارهای داده فشرده‌تری مانند درخت‌ها و یا درخت‌های باینری استفاده می‌کنند، معمولاً کارایی حافظه بالاتری دارند.

- زمان جستجو:

- * زمان لازم برای یافتن تطابق در جدول، معیار مهم دیگری است. در مسیریاب‌ها، این معیار باید تا حد ممکن کمینه باشد.
- * روش ارزیابی: می‌توان میانگین زمان جستجو یا پیچیدگی زمانی روش‌ها (مثلاً $O(\log n)$) یا $O(1)$ برای جستجوهای خاص را بررسی کرد. روش‌های سریع‌تر زمان جستجو را کاهش می‌دهند و عملکرد بهتری دارند.

ادامه پاسخ

- مقیاس پذیری:

- * روش باید بتواند با افزایش تعداد و طول آدرس ها (به خصوص در شبکه های بزرگ و پیچیده)، همچنان با سرعت و دقت بالا کار کند.
- * روش ارزیابی: عملکرد روش ها در حجم داده های بزرگ و همچنین میزان تأثیر افزایش تعداد آدرس ها بر زمان جستجو و حافظه مصرفی بررسی می شود.

- انعطاف پذیری و بروزرسانی:

- * روش باید در برابر تغییرات آدرس ها، افزودن یا حذف آن ها انعطاف پذیر باشد و امکان بروزرسانی سریع و آسان را فراهم کند.
- * روش ارزیابی: سرعت و سهولت عملیات درج، حذف و بروزرسانی آدرس ها در روش بررسی می شود. روش های با ساختار پویا (مانند ساختارهای درختی) معمولاً در این زمینه عملکرد بهتری دارند.

- دسته بندی روش های جستجوی آدرس IP در جدول جلورانی را با ذکر ویژگی های هر روش بیان کنید.

پاسخ

- روش های مبتنی بر درخت (Trie-Based Methods):

- * ویژگی ها: این روش ها از ساختار درختی مانند Binary Trie استفاده می کنند و آدرس های IP را به صورت بیتی ذخیره می کنند. به این صورت که هر سطح درخت نشان دهنده یک بیت از آدرس است و مسیر به پایین ترین سطح نشان دهنده طولانی ترین تطابق است.
- * مزایا: زمان جستجوی نسبتاً سریع و مقیاس پذیری خوب.
- * معایب: در شرایط خاص، ممکن است نیاز به حافظه زیادی داشته باشند و نگهداری و بروزرسانی آنها پیچیده باشد.

- روش های مبتنی بر هشینگ (Hashing-Based Methods):

- * ویژگی ها: از جدول های هش برای ذخیره پیشوندهای IP استفاده می کنند. در این روش، جستجو از طریق هش کردن انجام می شود.
- * مزایا: سرعت جستجوی بسیار سریع، به ویژه در شرایطی که جستجو با پیچیدگی $O(1)$ انجام شود.
- * معایب: مقیاس پذیری کمتر نسبت به روش های دیگر و محدودیت در پردازش طولانی ترین پیشوندها به دلیل نبود ساختار سلسله مراتبی.

- روش های مبتنی بر جستجوی باینری (Binary Search-Based Methods):

- * ویژگی ها: از جستجوی باینری برای یافتن طولانی ترین پیشوند استفاده می کنند. این روش به خصوص برای شبکه هایی با پیشوندهای کوتاه تر مناسب است.
- * مزایا: زمان جستجوی ثابت و مناسب در اندازه های کوچک و متوسط.
- * معایب: کارایی حافظه کمتر نسبت به سایر روش ها در شبکه های بزرگ.

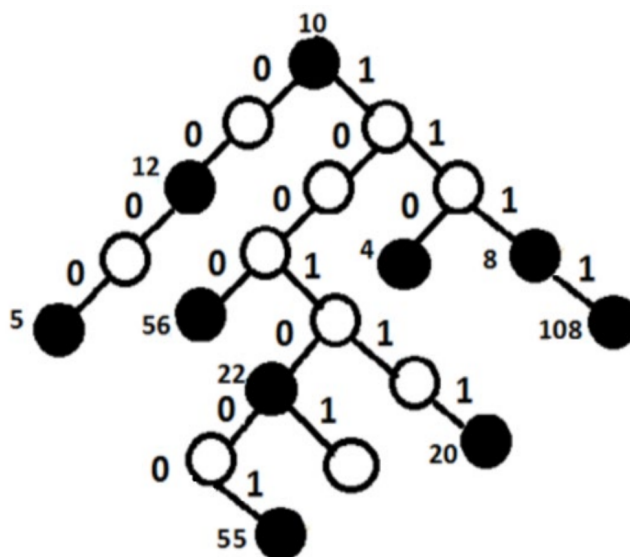
ادامه پاسخ

- روش‌های مبتنی بر جداول TCAM:

- * ویژگی‌ها: در این روش‌ها از حافظه TCAM استفاده می‌شود که امکان جستجوی همزمان چندین ورودی را فراهم می‌آورد.
- * مزایا: سرعت بسیار بالای جستجو و پشتیبانی از طولانی‌ترین پیشنود.
- * معایب: قیمت بالا و مصرف انرژی زیاد، همچنین نیاز به تنظیم دقیق برای بهینه‌سازی عملکرد.

سوال سوم

درخت باینری نشان داده شده در زیر، از روی جدول جلورانی در یک مسیریاب ایجاد شده است و جست و جو در جدول جلورانی برای پیدا کردن شماره پورت خروجی با پیمایش این درخت انجام می‌گردد. (در این درخت باینری، گره‌های مشکی، پیشوندهای آدرس جدول جلورانی هستند و مقادیر آن‌ها بیانگر شماره پورت خروجی است.)



شکل ۱: درخت باینری

۱. بسته‌ای با آدرس مقصد 90B28FF1 از کدام پورت خارج می‌گردد؟ (آدرس‌های مقصد در مبنای ۱۶ نمایش داده شده‌اند.)

پاسخ

بر اساس درخت داده شده، جدول مسیریابی به صورت زیر به دست می‌آید:

Prefix	Output Port
*	10
111*	8
110*	4
1000*	56
10010*	22
100100*	60
100111*	20
00*	12
0000*	5

اگر معادل باینری آدرس داده شده را بنویسیم:

$(0X90B28FF1) = 1001\ 0000\ 1011\ 0010\ 1000\ 1111\ 1111\ 0001$

می‌بینیم که براساس الگوریتم Longest Prefix Matching بیشترین Matching را با سطر ۶ جدول دارد. پس در نتیجه، این بسته از پورت شماره ۶۰ خارج می‌شود.

۲. بسته‌ای با آدرس مقصد A2AB11C3 از کدام پورت خارج می‌گردد؟

پاسخ

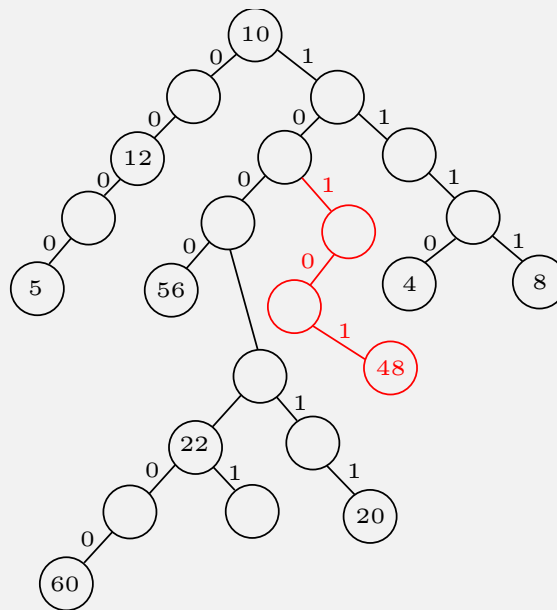
معادل باینری آدرس داده شده را می نویسیم:

(0XA2AB11C3) = 1010 0010 1010 1011 0001 0001 1100 0011

بیشترین Matching را با سطر ۱ جدول است. پس، این بسته از پورت شماره ۱۰ خارج می‌شود.

۳. آدرس پیشوندی *10101 با پورت خروجی ۵۰ را به درخت اضافه نمایید.

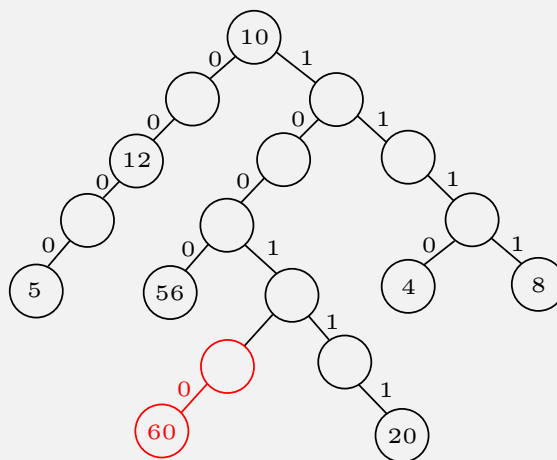
پاسخ



شکل ۲: درخت جدید

۴. آدرس پیشوندی *10010 با پورت خروجی ۲۲ را از درخت حذف نمایید.

پاسخ



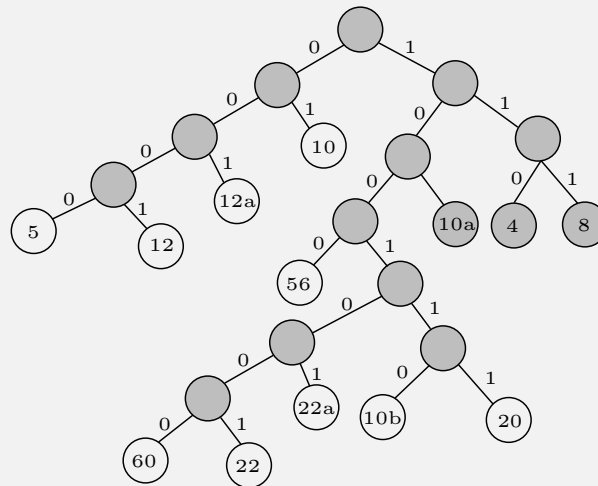
شکل ۳: درخت جدید

سوال چہارم

درخت سوال ۳ را به فرم disjoint-prefix binary trie تبدیل نمایید. حافظه مورد نیاز جهت نگهداری درخت سوال ۳ و سوال ۴ را مقایسه کنید.

پاسخ

درخت Disjoint حاصل به صورت زیر است:



شکل ۴: Disjoint درخت

به این درخت باید ۴ گره جدید دیگر اضافه شود تا درخت کامل شود. در اینجا مجموعاً ۱۳ برگ داریم و در این روش فقط برگ‌ها را در حافظه ذخیره می‌کنیم و در نتیجه کلاً اطلاعات ۱۳ برگ را در حافظه داریم. اما در سوال ۴ کل درخت که شامل ۱۹ گره است را باید در حافظه نگهداری کنیم. در نتیجه، در این روش، حافظه مصرفی کاهش می‌یابد.

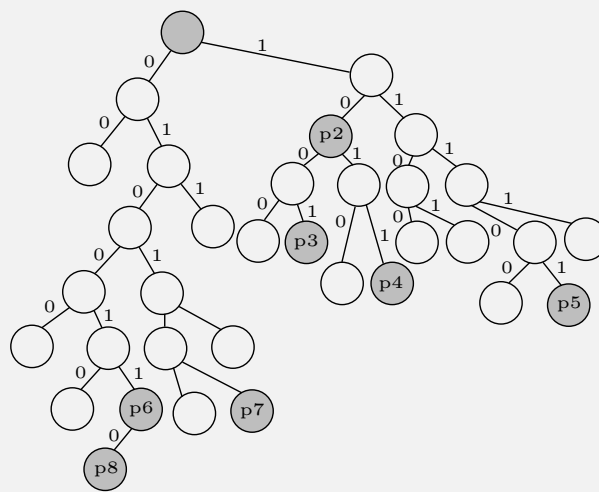
سوال پنجم

Path-compressed trie جدول جلورانی زیر را رسم کنید.

- P1 *
- P2 10*
- P3 1001*
- P4 1011*
- P5 11101*
- P6 010011*
- P7 010101*
- P8 0100110*

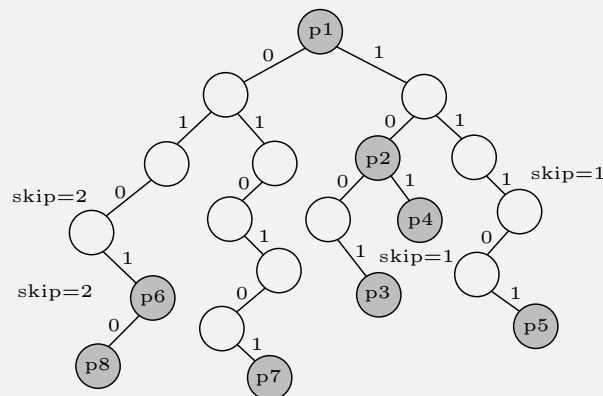
پاسخ

ابتدا Binary Trie جدول را به صورت زیر رسم می‌کنیم:



شکل ۵: درخت Binary

سپس درخت Path Compress را به صورت زیر رسم می‌کنیم:



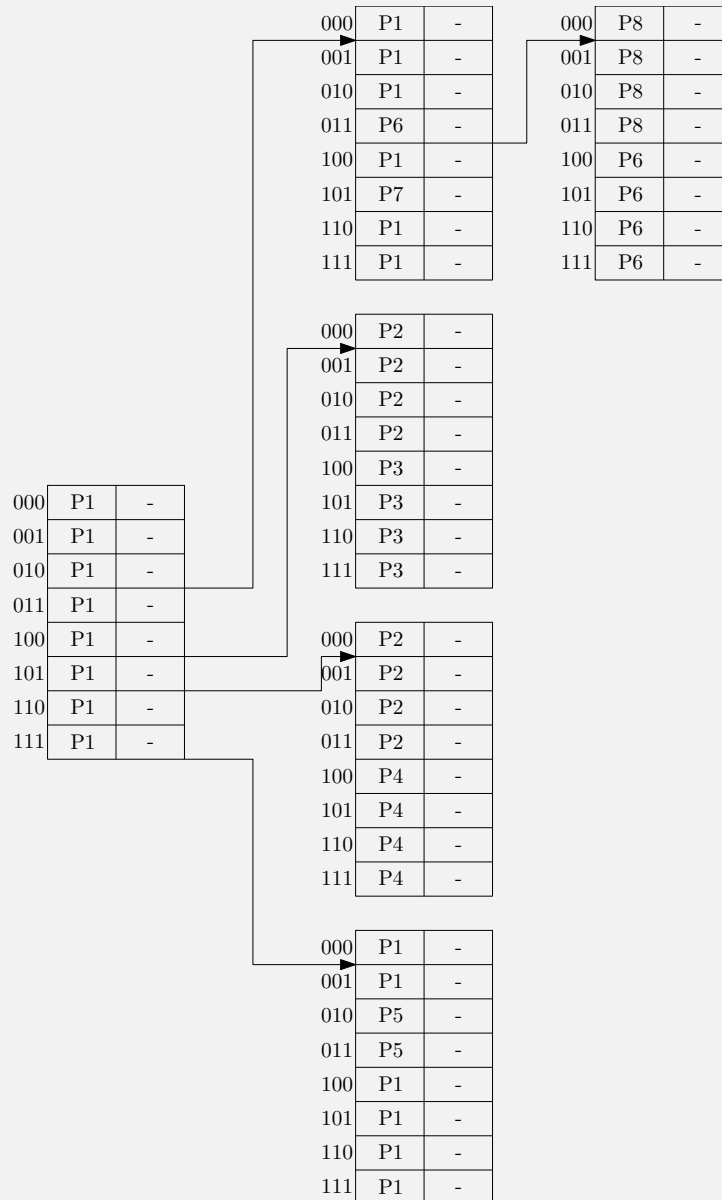
شکل ۶: درخت Path Compress

سوال ششم

Multi Bit Trie سه بیتی جدول مسیریابی سوال ۵ را رسم نموده و سپس درخت به دست آمده را به disjoint-prefix تبدیل نمایید.

پاسخ

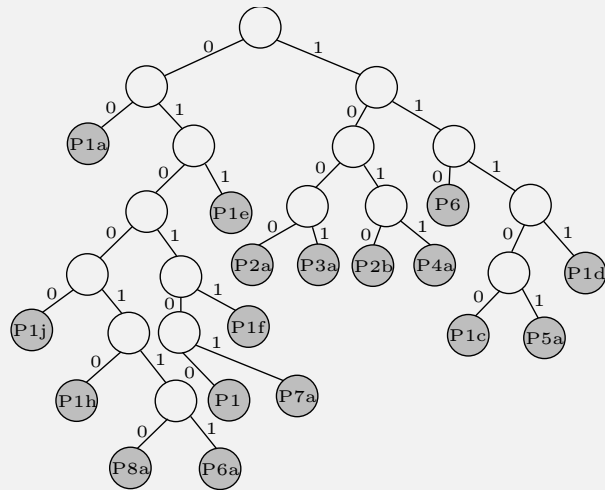
درخت Multi Bit Trie به صورت زیر رسم می شود:



شکل ۷: درخت Multi Bit Trie

و درخت Disjoint-Prefix را به صورت زیر رسم می کنیم:

پاسخ



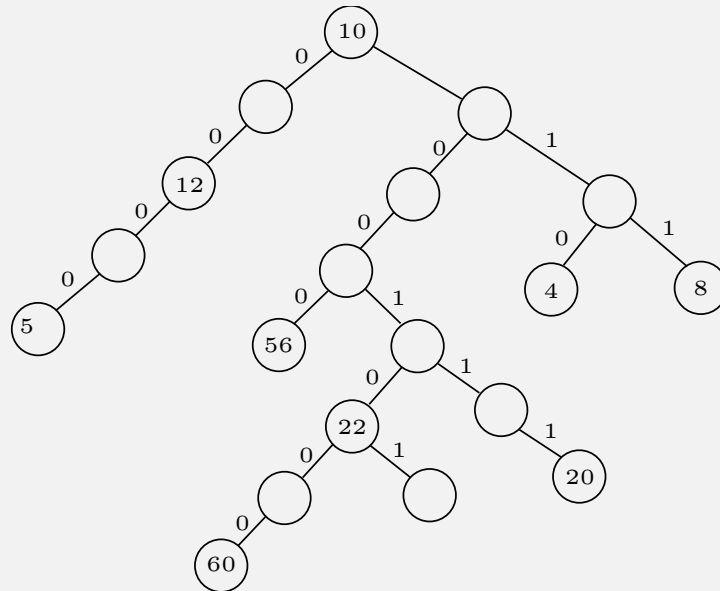
شکل ۸: درخت Disjoint-Prefix

سوال هفتم

LC-trie نظیر درخت سوال ۳ را رسم نمایید. رابطه تعداد branch ها و stride ها چیست؟

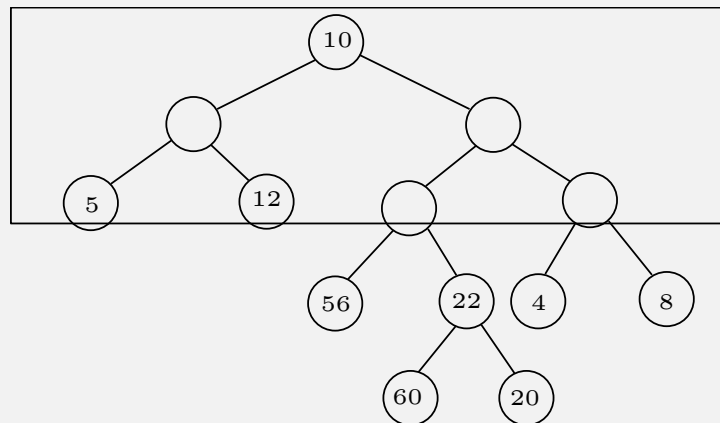
پاسخ

ابتدا درخت One Bit Trie را به صورت زیر رسم می‌کنیم:



شکل ۹: درخت One Bit Trie

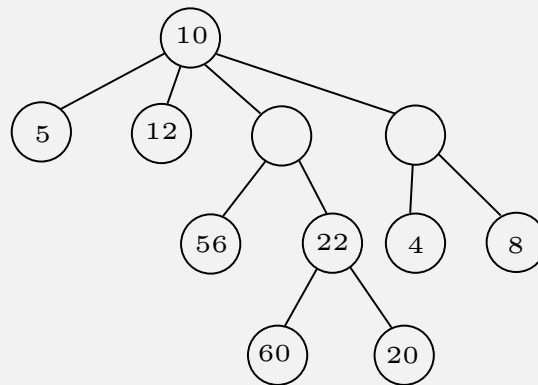
درخت Path Compressed Trie به صورت زیر می‌شود:



شکل ۱۰: درخت Path Compressed Trie

پاسخ

و در نهایت LC-Trie به صورت زیر رسم می شود:



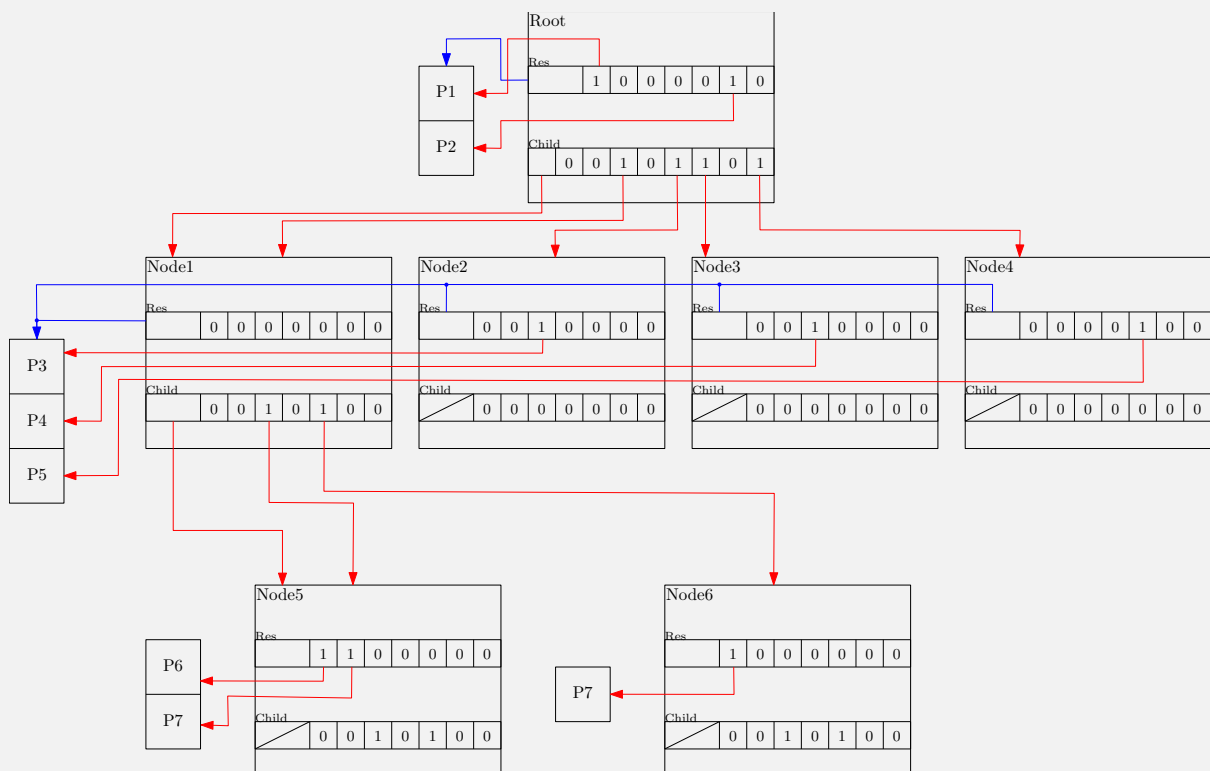
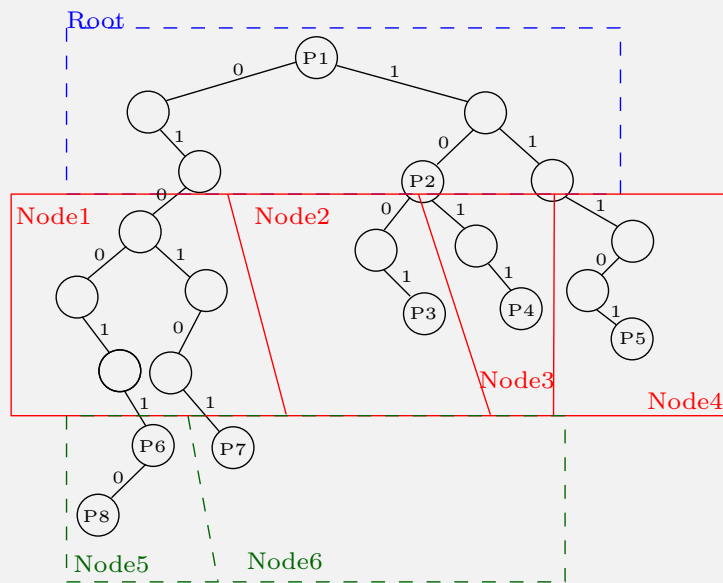
شکل ۱۱: درخت LC-Trie

مقدار Stride با تعداد Branch رابطه مستقیم دارد.
هرچقدر که مقدار Stride بیشتر باشد، تعداد Branch ها بیشتر می شود.

سوال هشتم

برای جدول مسیریابی سوال ۵، بر اساس روش Tree Bitmap ساختار داده را به طور کامل رسم نمایید. (Stride=3)

پاسخ



سوال نهم

در الگوریتم Binary search on prefix range در صورتی که n داده، m بیتی داشته باشیم و از k -way search استفاده کنیم.

- در این الگوریتم چه ارتباطی بین n ، m و k وجود دارد؟

پاسخ

در هر جست و جو، $\frac{1}{k}$ از جدول مرحله بعدی جست و جو باقی می ماند زیرا باید، $\log(k)$ بیت را در جدول جست و جو کنیم. جدولی $2n$ ردیفی داریم که دارای m بیت در هر ردیف است.

- در بدترین حالت مرتبه زمانی جست و جو و حافظه مورد نیاز را محاسبه نمایید.

پاسخ

در بدترین حالت، پیچیدگی حافظه و زمان به ترتیب برابر هستند با $O(N)$ و $O(\log(n))$

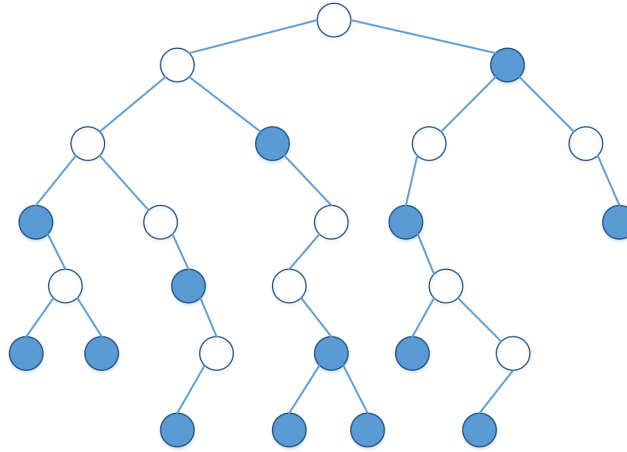
- در صورتی که عمل جست و جو توسط پردازنده ای با فرکانس ساعت 3.2 MHz و بر روی یک جدول جلورانی با 90000 سطر (entries) انجام شود، بیشترین زمان مورد نیاز و متوسط زمان مورد نیاز و میزان حافظه مورد نیاز را محاسبه نمایید.

پاسخ

با فرض اینکه هر بار جدول به k قسمت تقسیم شود و سپس مقایسه انجام شود، هر دفعه k دسترسی به حافظه خواهیم داشت.

سوال دهم

یکی از روش‌های کاهش هزینه توان مصرفی در TCAM روش Trie-Based Table Partitioning است. با اعمال این دو روش بر روی درخت زیر، جدولی مشابه جدول صفحه ۶۴ کتاب بدست آورید. گره‌های آبی رنگ شامل آدرس‌های Prefix هستند. (مقدار اندازه بلوک‌های حافظه را ۴ در نظر بگیرید؛ $b = 4$)

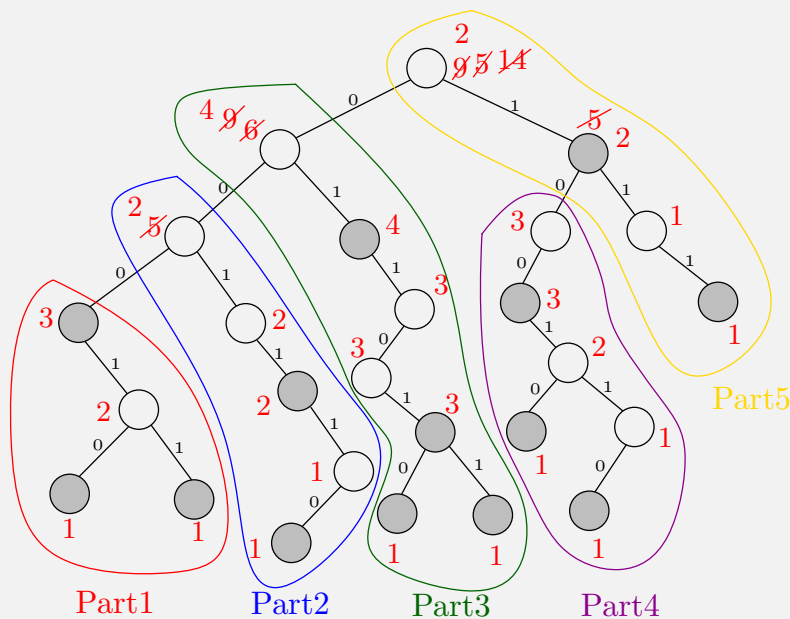


شکل ۱۲: درخت باینری

پاسخ

مقدار Value را برای هر گره می‌نویسیم و در هر مرحله که تکه‌ای از درخت را جدا می‌کنیم، این مقدار را آپدیت می‌کنیم و مقدار و مقدار آن را با رنگ قرمز در کنار هر Node مشخص می‌کنیم. باید $2 \leq Value \leq 4$ را جدا کنیم.

۱. Preorder



شکل ۱۳: Preorder

[illegible]

Prefix	Bucket Prefixes	Bucket Size	Covering Prefix
0001*,001*	00010*,00011*,0011*,001110*	4	0001*,0011*
01*	01*,01101*,011010*,011011*	4	01*,01101*
10*,11*	100*,10010*,100110*,111*	4	100*,111*
*	000*,1*	2	1*,000*