



دانشگاه پیام نور  
دانشکده فنی و مهندسی

پایان نامه

برای دریافت درجه کارشناسی ارشد  
رشته مهندسی کامپیوتر - گرایش نرم افزار  
گروه فناوری اطلاعات و ارتباطات

## **پیشنهاد یک الگوریتم موازی و آگاه از حافظه نهان برای ساخت درخت پسوندی مربوط به دنباله های DNA**

معصومه السادات علوی

استاد راهنما:

جناب آقای دکتر معظم

استاد مشاور:

جناب آقای دکتر سید علی رضوی

شهریور ۱۳۹۱



دانشگاه پیام نور  
دانشکده فنی و مهندسی  
دانشگاه پیام نور مرکز تهران

پایان نامه  
برای دریافت درجه کارشناسی ارشد  
رشته مهندسی کامپیوتر - گرایش نرم افزار  
گروه فناوری اطلاعات و ارتباطات

## **پیشنهاد یک الگوریتم موازی و آگاه از حافظه نهان برای ساخت درخت پسوندی مربوط به دنباله‌های DNA**

معصومه السادات علوی

استاد راهنما:

جناب آقای دکتر معظم

استاد مشاور:

جناب آقای دکتر سید علی رضوی

شهریور ۱۳۹۱

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

تاریخ .....  
شماره .....  
پیوست .....



دانشگاه پیام نور

دانشگاه پیام نور استان تهران



جمهوری اسلامی ایران  
وزارت علوم، تحقیقات و فناوری

مرکز شمیرانات

### تصویب نامه

بایان نامه کارشناسی ارشد رشته مهندسی کامپیوتر (نرم افزار)

تحت عنوان:

"پیشنهاد یک الگوریتم موازی و آگاه از کش (cash) برای ساخت درخت

پسوندی مربوط به دنباله های DNA"

ساعت: ۱۱-۱۲

تاریخ دفاع: ۱۳۹۱/۰۶/۳۰

درجه ارزشیابی: عالی

نمره: ۱۹/۱

هیات داوران:

امضاء	مرتبه علمی	نام و نام خانوادگی	داوران
	استاد	دکتر محمدهادی معظم	استاد راهنما
	استاد	دکتر سید علی رضوی ابراهیمی	استاد مشاور
	استاد	دکتر مهدی جوانمرد	استاد داور
	استاد	دکتر مهدی جوانمرد	نماینده گروه

تهران-بزرگراه ارتش-انتهای  
بلوار شهید مژدی (اوشان)  
خیابان شهید پیروز شفیعی  
خیابان یاران-خیابان یاران دوم  
دانشگاه پیام نور مرکز شمیرانات

تلفن: ۴-۲۲۱۹۵۳۰۳

دورنگار: ۴-۲۲۴۸۴۸۳۴

<http://teh-shemiranat.pnu.ac.ir>

[info@fani.pnu.ac.ir](mailto:info@fani.pnu.ac.ir)

تقدیم به:

مادر فداکارم که مهر بیکرانیش را نمی توانم جبران کنم.  
پدر عزیزم که همیشه مدیون محبت هایش هستم.  
همسر مهربانم که یاری هایش تکیه گاهم در مراحل زندگی است.

## تشکر و قدردانی:

در ابتدا پروردگار را به خاطر لطف و توجهی که در مراحل مختلف زندگی شامل حالم نموده شاکرم. به عنوان وظیفه، لازم است از استادان و دوستانی که در تدوین این تحقیق مرا یاری نموده‌اند تشکر نمایم.

از استاد راهنمای گران‌قدر جناب آقای دکتر معظم که همواره از توصیه‌های ایشان بهره‌مند شدم کمال تشکر را دارم. همچنین جا دارد از زحمات استاد مشاور محترم جناب آقای دکتر رضوی سپاسگزاری نمایم. بعلاوه توفیق آن را داشته‌ام که از نظرات جناب آقای دکتر پارسا، برخوردار شوم که از صمیم قلب از ایشان سپاسگزاری می‌نمایم. نیز از آقای دکتر حمزه‌ای به عنوان نخستین الهام بخش در انتخاب موضوع ساخت درخت‌های پسوندی که زمینه‌ساز این تحقیق شده است، قدردانی نمایم. همچنین از آقای مهندس کهندانی که در پیاده‌سازی این تحقیق کمک بسیار بزرگی بوده‌اند، کمال تشکر را دارم.

از حمایت بی‌دریغ تک‌تک افراد خانواده که تدوین این تحقیق را برایم فراهم نموده‌اند نهایت سپاسگزاری را دارم.

## چکیده

درخت پسوندی ساختار داده‌ای است که تمامی پسوندهای یک رشته را نمایش می‌دهد. به کمک تشکیل این ساختار داده‌ای می‌توان اعمالی نظیر یافتن یک زیر رشته را در دنباله‌های طولانی مانند DNA به طور کلی در دنباله‌ها انجام داد. الگوریتم‌های متفاوتی برای ساخت درخت پسوندی وجود دارد. الگوریتم‌های اولیه، تشکیل درخت پسوندی را در زمان خطی ممکن می‌ساخت، اما برای دنباله‌های طولانی‌تر این الگوریتم‌ها دارای کارایی لازم نیستند چرا که دنباله ورودی و درخت حاصل در حافظه اصلی جای نمی‌گیرند و در نتیجه دست‌یابی‌هایی به دیسک وجود خواهد داشت. بنابراین تمرکز الگوریتم‌های بعدی در کاهش این دست‌یابی‌ها از طریق ارائه روش‌هایی برای جای دادن درخت پسوندی حاصل در حافظه بوده است.

در این تحقیق الگوریتم موازی برای ساخت درخت پسوندی ارائه شده است. این الگوریتم درخت پسوندی را به جای حافظه اصلی در کش جای می‌دهد. همچنین موازی بودن آن امکان اجرای سریع‌تر الگوریتم را ممکن می‌سازد.

**واژه‌های کلیدی:** درخت پسوندی، اندیس‌گذاری دنباله‌ها، الگوریتم موازی.

فصل ۱ مقدمه ۱

۱-۱. مقدمه ..... ۲

۱-۲. تعریف مسئله و سؤالات اصلی تحقیق ..... ۶

۱-۳. فرضیه‌ها ..... ۷

۱-۴. اهداف تحقیق ..... ۸

۱-۵. روش تحقیق ..... ۹

۱-۶. مراحل انجام تحقیق ..... ۹

۱-۷. ساختار پایان‌نامه ..... ۱۰

فصل ۲ تعاریف اولیه ۱۱

۲-۱. تاریخچه درخت پسوندی ..... ۱۲

۲-۲. آشنایی با Trie ..... ۱۲

۲-۳. درخت پسوندی و تعاریف پایه ..... ۱۳

۲-۴. معرفی یک الگوریتم ابتدایی ..... ۱۶

۲-۵. ذخیره‌سازی درخت پسوندی - مسائل مربوط به پیاده‌سازی ..... ۱۷

فصل ۳ معرفی و بررسی الگوریتم‌های مبتنی بر حافظه ۲۴

۳-۱. الگوریتم Ukkonen و ساخت آنلاین درخت پسوندی ..... ۲۵

۳-۲. الگوریتم WOTD ..... ۳۱

۳-۳. الگوریتم M و کاهش حافظه مورد نیاز ..... ۳۴

۳-۴. الگوریتم Farach ..... ۴۰

۳-۵. جمع‌بندی ..... ۴۶

فصل ۴ معرفی و بررسی الگوریتم‌های مبتنی بر دیسک ۴۹

۴-۱. الگوریتم‌های In-Core String ..... ۵۱

۴-۱-۱. الگوریتم Hunt ..... ۵۲

۴-۱-۲. الگوریتم DYNACLUSTER ..... ۵۵

۴-۱-۳. روش Top-Q برای کاهش هزینه‌های I/O ..... ۶۳

۴-۱-۴. روش TDD، ساخت درون دیسک درخت پسوندی به صورت بالا - به - پایین ..... ۶۵

۴-۱-۵. روش پارتیشن‌بندی و ادغام الگوریتم Trellis ..... ۶۹



۷۴	..... ۶-۱-۴ جمع‌بندی
۷۷	..... ۲-۴. الگوریتم‌های Out-of-String
۷۸	..... ۱-۲-۴. الگوریتم ST-MERGE
۸۲	..... ۲-۲-۴. الگوریتم Wavefront
۸۹	..... ۳-۲-۴. الگوریتم DiGeST
۹۳	..... ۴-۲-۴. جمع‌بندی

## ۹۴ فصل ۵ معرفی و بررسی الگوریتم‌های موازی

۹۵	..... ۱-۵. الگوریتم PWAVEFRONT
۹۸	..... ۲-۵. الگوریتم CMPUTree

## ۱۰۱ فصل ۶ معرفی روش پیشنهادی

۱۰۲	..... ۱-۶. الگوریتم موازی و آگاه از حافظه نهان برای ساخت درخت پسوندی مربوط به دنباله‌های DNA
۱۰۳	..... ۲-۶. شرح الگوریتم
۱۰۴	..... ۱-۲-۶. مرحله اول: استخراج مجموعه پیشنوندهای با طول متغیر
۱۰۷	..... ۲-۲-۶. مرحله دوم: یافتن مکان پسوندهای هر پیشنهاد
۱۰۸	..... ۳-۲-۶. مرحله سوم: پیش‌پردازش جهت مرتب‌سازی پسوندها
۱۰۹	..... ۴-۲-۶. مرحله چهارم: ساخت زیردرخت مربوط به هر پیشنهاد
۱۱۳	..... ۳-۶. جمع‌بندی

## ۱۱۵ فصل ۷ نتایج تجربی

۱۱۶	..... ۱-۷. پارامترهای پیاده‌سازی
۱۱۷	..... ۲-۷. نتایج پیاده‌سازی
۱۱۷	..... ۱-۲-۷. ارزیابی روش پیشنهادی و مقایسه با الگوریتم CMPUTree
۱۲۰	..... ۲-۲-۷. مقایسه روش پیشنهادی با الگوریتم‌های پیشین ساخت درخت پسوندی
۱۲۲	..... ۳-۷. جمع‌بندی

## ۱۲۴ فصل ۸ جمع‌بندی و کارهای آینده

۱۲۶	..... ۱-۸. انگیزه تحقیق
۱۲۷	..... ۲-۸. کارهای آینده

۱۲۸

مراجع

۱۳۱

واژه‌نامه

عنوان	صفحه
فصل ۱ مقدمه	۱
شکل ۱-۱. رشد نمایی اندازه GenBank .....	۳
شکل ۱-۲. درخت پسوندی برای رشته TATAT .....	۴
شکل ۱-۳. یافتن الگوی ATA در رشته TATAT .....	۵
شکل ۱-۴. یافتن طولانی‌ترین زیر رشته مشترک برای رشته TATAT و AATAA .....	۶
<b>فصل ۲ تعاریف اولیه</b>	
شکل ۲-۱. ساختار trie برای دو رشته ab و abc .....	۱۳
شکل ۲-۲. ساختار trie برای دو رشته \$abc و \$ab .....	۱۳
شکل ۲-۳. ساختار trie فشرده شده شکل ۲-۲ .....	۱۳
شکل ۲-۴. درخت پسوندی برای رشته TATAT .....	۱۴
شکل ۲-۵. نمایش زنجیره‌های پیوندی درخت حاصل از رشته TATAT .....	۱۵
شکل ۲-۶. الگوریتم ابتدایی برای ساخت درخت پسوندی .....	۱۷
شکل ۲-۷. ساخت مرحله به مرحله درخت پسوندی، به کمک الگوریتم ابتدایی برای رشته .....	۱۸
شکل ۲-۸. نمایش درخت پسوندی با برچسب‌های بهینه‌شده .....	۱۹
شکل ۲-۹. نمایش آرایه‌ای برای رشته \$TATAT. الفبا از سه حرف A، T و \$ تشکیل شده است و آرایه مربوط به هر گره از ۳ سلول تشکیل شده است .....	۲۰
شکل ۲-۱۰. الف) نمایش فرزند چپ - همزاد راست برای رشته TATAT. هر گره دو اشاره‌گر یکی به فرزند چپ و یکی به همزاد راست دارد .....	۲۱
شکل ۲-۱۱. نمایش همزادها بصورت پشت سرهم در حافظه. سمبل * نشان‌دهنده همزاد آخر می‌باشد. هر گره تنها اشاره‌گر به چپ‌ترین فرزند و اندیس شروع ساق ورودی را نگهداری می‌کند .....	۲۲
<b>فصل ۳ معرفی و بررسی الگوریتم‌های مبتنی بر حافظه</b>	
شکل ۳-۱. ساخت مرحله ۲ تا ۶ ساختار Trie به کمک الگوریتم Ukkonen برای نمایش S=mississippi .....	۲۹
شکل ۳-۲. گره فعال و گره پایانی برای T <sub>4</sub> و T <sub>5</sub> . دایره نشان‌دهنده گره فعال و مربع نشان‌دهنده گره پایانی می‌باشد .....	۳۰
شکل ۳-۳. شکستن ساق در حین ساخت درخت پسوندی (الف) درخت قبل از شکستن ساق (ب) درخت پس از شکستن و قبل از به‌روزرسانی (ج) درخت پس از به‌روزرسانی .....	۳۱
شکل ۳-۴. ساخت ۹ مرحله اول درخت پسوندی برای رشته mississippi به کمک الگوریتم Ukkonen .....	۳۲
شکل ۳-۵. الف) درخت پسوندی نهایی برای رشته \$mississippi به کمک الگوریتم Ukkonen .....	۳۲

۳۲ ..... (ب) همان درخت با فشرده‌سازی برچسب ساق‌ها  
**شکل ۳-۶**. درخت حاصل از اعمال الگوریتم WOTD روی رشته \$mississippi\$. (الف) مرتب شدن پسوندهای گروه i و اضافه شدن LCP اعضا به درخت. (ب) تشکیل ۴ گروه جدید و ایجاد دو گره برگ برای گروه \$ و p (ج) ایجاد یک گره میانی برای گروه s در مرحله قبل (د) اتمام عملیات اضافه شدن پسوندهایی که با i شروع می‌شوند ..... ۳۵  
**شکل ۳-۷**. ۶ مرحله اول ساخت درخت پسوندی به کمک الگوریتم M برای رشته \$mississippi ..... ۳۷  
**شکل ۳-۸**. نمایش برچسب ساق‌ها به کمک اندیس شروع و پایان برچسب برای درخت پسوندی رشته \$mississippi ..... ۳۹  
**شکل ۳-۹**. اضافه شدن S4 در الگوریتم M، با استفاده از زنجیره‌های پسوندی ..... ۳۹  
**شکل ۳-۱۰**. زیر درخت پسوندی فرد و زوج برای رشته \$mississippi\$ (الف) زیردرخت فرد (ب) زیردرخت زوج ۴۱  
**شکل ۳-۱۱**. ساخت درخت پسوندی فرد برای رشته \$S = 21441441331\$ (الف) درخت پسوندی برای \$S' = 362541\$ (ب) زیر درخت فرد اولیه (ج) زیردرخت فرد نهایی ..... ۴۲  
**شکل ۳-۱۲**. الگوریتم ساخت درخت پسوندی از LCA دو گره مجاور ..... ۴۴  
**شکل ۳-۱۳**. ادغام زیردرخت پسوندی زوج و فرد (الف) زیر درخت پسوندی فرد (ب) زیر درخت پسوندی زوج (ج) درخت ادغام شده اولیه (د) درخت پسوندی نهایی پس از اعمال اصلاحات ادغام ..... ۴۵

## فصل ۴ معرفی و بررسی الگوریتم‌های مبتنی بر دیسک ۴۹

**شکل ۴-۱**. روش بدیهی برای ساخت درخت پسوندی ..... ۵۳  
**شکل ۴-۲**. مراحل ساخت زیردرخت برای پیشوند s و رشته ورودی \$S = mississippi\$ به کمک الگوریتم Hunt ..... ۵۵  
**شکل ۴-۳**. اضافه شدن گره‌ها به درخت پسوندی با الگوریتم Hunt ..... ۵۸  
**شکل ۴-۴**. تأثیر انتخاب مقدار t در رشته کروموزوم ۱۸ با استفاده از 16MB حافظه اصلی ..... ۵۹  
**شکل ۴-۵**. ساخت درخت پسوندی به کمک الگوریتم DYNACLUSTER. (الف) ایجاد پارتیشن ریشه (ب) ایجاد پارتیشن برگ (ج) ایجاد پارتیشن میانی (د) ایجاد پارتیشن برگ ..... ۶۲  
**شکل ۴-۶**. بررسی تعداد دسترسی به یک گره برای رشته کروموزوم واقعی ..... ۶۳  
**شکل ۴-۷**. مراحل الگوریتم PWOTD روی رشته ورودی \$S = mississippi\$ برای ایجاد زیر درخت مربوط به پارتیشن i ..... ۶۷  
**شکل ۴-۸**. مرور کلی الگوریتم Trellis ..... ۶۹  
**شکل ۴-۹**. مرحله ادغام در Trellis (الف) برچسب ساق‌های خروجی، سمبل آغازین یکسان ندارد (ب) برچسب ساق‌ها، پیشوند مشترک دارند ..... ۷۱  
**شکل ۴-۱۰**. مرحله الگوریتم Trellis روی رشته ورودی \$S = mississippi\$ (الف) ساخت زیر درخت پسوندی برای رشته \$S1 = mississippi\$ (ب) ساخت زیر درخت پسوندی برای رشته \$S2 = ssippi\$ ..... ۷۳  
**شکل ۴-۱۱**. مرور کلی الگوریتم ST-MERGE ..... ۷۹  
**شکل ۴-۱۲**. نمونه‌ای از مرحله ادغام در الگوریتم ST-MERGE. (الف) سه زیردرخت T1، T2 و T3 که باید ادغام شوند و درخت حاصل از ادغام می‌باشد. مثلث‌ها زیر یک گره، نشان‌دهنده زیردرخت زیرین یک گره می‌باشد. (ب) ادغام ساق‌های گروه A (ج) ادغام ساق‌های گروه G (د) ادغام ساق‌های گروه T ..... ۸۱  
**شکل ۴-۱۳**. بازیابی زنجیره‌های پسوندی در الگوریتم WAVEFRONT ..... ۸۵  
**شکل ۴-۱۴**. ساخت مرحله به مرحله زیردرخت مربوط به is در رشته \$S = mississimissis\$ به کمک الگوریتم WAVEFRONT. در این نمونه فرض شده است که B برابر ۵ کاراکنتر است؛ پس ۳ بلوک وجود خواهد داشت. (الف) پسوند در محل ۱ به صورت کامل اضافه شده است. پسوند در محل ۴ بصورت نیمه‌تمام اضافه شده است، وضعیت فعلی نگهداری می‌شود. (ب) پسوند در محل ۴ بصورت کامل اضافه شده است. پسوند در محل ۹ بصورت نیمه‌تمام اضافه شده است. (ج) پسوند در محل ۱۲ بصورت کامل اضافه شده است. پسوند در محل ۹ همچنان بصورت نیمه‌تمام

باقی مانده است. (د) بارگذاری B1 و B1 تغییری در زیردرخت حاصل نمی‌کند و در این مرحله B1 و B2 بارگذاری می‌شوند. سپس پسوند در محل ۹ به زیردرخت اضافه می‌شود. تمامی پسوندها پردازش شده‌اند و نیازی به بارگذاری B2 در حافظه نمی‌باشد.....  
 شکل ۴-۱۵. ساخت مرحله به مرحله درخت پسوندی به کمک الگوریتم DiGeST.....  
 ۸۷ .....  
 ۹۲ .....

۹۴

## فصل ۵ معرفی الگوریتم‌های موازی

۱۰۱

### فصل ۶ معرفی روش پیشنهادی

شکل ۶-۱. شبه کد مربوط به مرحله یافتن مجموعه پارتیشن‌ها. تابع چهار متغیر  $B_c$ , temp,  $l$  و  $th$  را بعنوان ورودی می‌پذیرد و مجموعه پارتیشن‌ها و مجموعه پیشوندهای را خروجی می‌دهد.....  
 شکل ۶-۲. ایجاد مجموعه پیشوندها برای رشته  $S = \text{mississippi}$  با  $|S| = 4$ . عملیات گسترش برای پیشوند  $i$  و پیشوند  $s$  انجام شده است. مجموعه نهایی پارتیشن‌ها در سمت راست نشان داده شده است.....  
 شکل ۶-۳. ایجاد زیر درخت پسوندی برای پارتیشن مربوط به  $i$  در رشته  $\text{mississippi}$ . برچسب ساق‌ها، با اندیس شروع و پایان نشان داده شده است.....  
 ۱۰۵.....  
 ۱۰۷.....  
 ۱۱۳.....

۱۱۵

### فصل ۷ نتایج تجربی

شکل ۷-۱. زمان اجرای دو الگوریتم PCST و CMPUTree روی ژنوم انسان.....  
 ۱۱۸.....

۱۲۴

### فصل ۸ نتیجه‌گیری و کارهای آینده

۱۲۸

مراجع

۱۳۱

واژه‌نامه

صفحه	عنوان
۱	فصل ۱ مقدمه
۱۱	فصل ۲ تعاریف اولیه
۲۴	فصل ۳ معرفی الگوریتم‌های مبتنی بر حافظه
۴۹	فصل ۴ معرفی الگوریتم‌های مبتنی بر دیسک
۹۴	فصل ۵ معرفی الگوریتم‌های موازی
۱۰۱	فصل ۶ معرفی روش پیشنهادی
۱۱۱	جدول ۶-۱. پسوندهای مرتب شده برای پارتیشن i در رشته mississippi\$
۱۱۱	جدول ۶-۲. یافتن آرایه LCP مربوط به پارتیشن i و رشته mississippi\$
۱۱۵	فصل ۷ نتایج تجربی
۱۱۷	شکل ۷-۱. مجموعه داده‌های استفاده شده برای بدست آوردن نتایج حاصل از الگوریتم
۱۲۱	شکل ۷-۲. میزان حافظه استفاده شده در PCST
۱۲۳	شکل ۷-۳. مقایسه الگوریتم‌های موجود برای ساخت درخت پسوندی
۱۲۴	فصل ۸ نتیجه‌گیری و کارهای آینده
۱۲۸	مراجع
۱۳۱	واژه‌نامه

## فهرست علائم اختصاری

---

DFS	Depth First Search	جستجوی اول-عمق
LCA	Least Common Ancestor	نزدیک‌ترین جد مشترک
LCP	Longest Common Prefix	طولانی‌ترین پیشوند مشترک
$p(v)$	$\text{parent}(v)$	والد گره‌ای به نام $v$
PCST	Parallel and Cache-Aware Suffix Tree Construction	ساخت موازی و آگاه از حافظه نهان درخت پسوندی
SL	Suffix Link	زنجیره پسوندی

# فصل ۱

## مقدمه



## ۱-۱. مقدمه

بیوانفورماتیک یا تحقیقات زیستی، دانش استفاده از علوم کامپیوتر در شاخه زیست‌شناسی مولکولی است. استفاده عمده از این واژه حداقل تا اواخر دهه ۱۹۸۰ در ژنتیک و مخصوصاً در حوزه‌هایی از ژنتیک شامل توالی DNA (تشخیص ترتیب نوکلئوتیدها A، G، C و T) بوده است (Bioinformatics, 2011). پروژه‌های مربوط به حوزه ژنتیک بسیار رایج است، بعنوان نمونه، پروژه شناسایی ژنوم انسان (وانگ<sup>۱</sup> و گوو<sup>۲</sup>، ۲۰۰۱) در سال ۱۹۹۰ آغاز شد و در سال ۲۰۰۳ پایان یافت و اکنون اطلاعات کاملی مربوط به توالی همه کروموزوم‌های انسان موجود می‌باشد (2011، بیوانفورماتیک).

امروزه بیوانفورماتیک شامل ایجاد و بهبود پایگاه داده‌ها و الگوریتم‌های ناشی از تحلیل داده‌های زیستی می‌شود. توسعه سریع فناوری‌های مربوط به ژنتیک در دهه‌های اخیر، سبب ایجاد میزان زیادی از اطلاعات درباره ژنوم (محتوای ژنتیکی) انسان‌ها، حیوانات و گیاهان و در نتیجه ایجاد پایگاه داده‌های بزرگ در این حوزه شده است. به طوری که اندازه پایگاه داده مربوط به ژن‌ها (GenBank) هر ۱۸ ماه دو برابر می‌شود (GenBank release notes, 2011). رشد نمایی اندازه این پایگاه داده در شکل ۱-۱ نشان داده شده است.

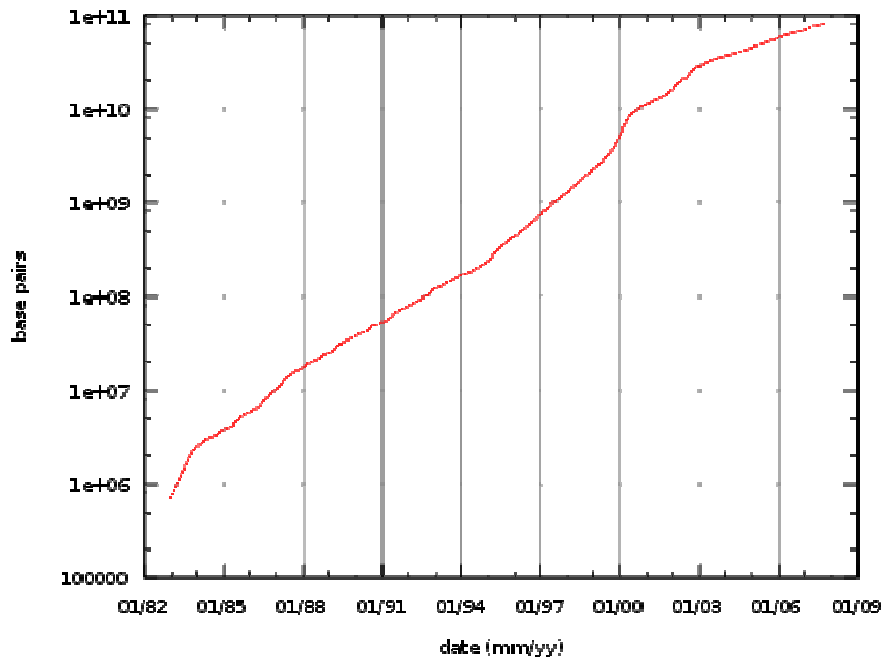
با توجه به آنچه درباره حجم پایگاه داده حاوی ژنوم موجودات زنده بیان شد، به راحتی می‌توان فهمید که اعمالی مانند جستجو در این پایگاه داده‌ها به موضوعی بحرانی تبدیل می‌شود. بنابراین برای استفاده کارا از این پایگاه داده‌ها، بکارگیری الگوریتم‌های کارآمد امری ضروری است. در نتیجه اندیس‌گذاری دنباله‌های موجود در این پایگاه داده‌ها، به طراحی الگوریتم‌های کارآمد در حل مسئله در ۷ کمک خواهد کرد.

نکته‌ای که باید به آن در اینجا اشاره شود این است که توالی DNA یک دنباله است و نه یک رشته؛ چیزی که این دو را از یکدیگر متمایز می‌کند این است که ممکن است در رشته

---

<sup>1</sup> Wang, J.

<sup>2</sup> Gu, J.



شکل ۱-۱. رشد نمایی اندازه GenBank

فاصله‌هایی وجود داشته باشد، این در حالی است که در دنباله‌ها یک چنین فاصله‌هایی وجود ندارد و تمامی حروف الفبا پشت سر هم قرار می‌گیرند. بنابراین ساختار DNA به گونه‌ای نیست که بتوان آن را بصورت کارا به کلمه‌ها شکست و در نتیجه روش‌های اندیس‌گذاری رشته‌های عادی مانند اندیس وارونه<sup>۱</sup> و B-Tree نمی‌تواند بصورت موثر در مورد دنباله‌های DNA بکار گرفته شود (با رس کی<sup>۲</sup>، استج<sup>۳</sup>، تومو<sup>۴</sup> و آپتون<sup>۵</sup>، ۲۰۰۸)

درخت پسوندی ساختار داده‌ای است که به کمک آن می‌توان تمامی پسوندهای یک رشته را نمایش داد، در اینصورت اعمالی نظیر یافتن یک الگوی خاص در دنباله‌های DNA به راحتی امکان‌پذیر می‌شود. جهت سهولت نوشتار در این تحقیق از تفاوت دنباله و رشته صرف‌نظر می‌شود، از رشته بجای دنباله استفاده می‌شود.

درخت پسوندی برای یک رشته، درخت ریشه‌داری است که برچسب ساق‌ها نمایانگر زیر

<sup>1</sup> Inverted Index

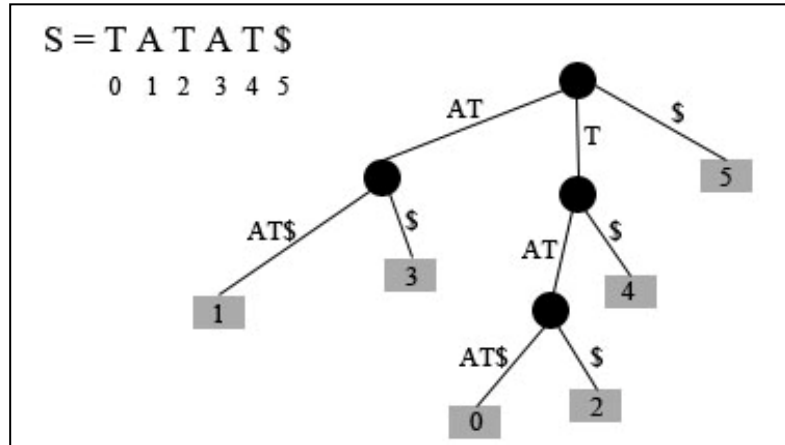
<sup>2</sup> Barsky, M

<sup>3</sup> Stege

<sup>4</sup> Thomo

<sup>5</sup> Upton

رشته‌ای از رشته اصلی می‌باشند. برای هر پسوند، یک مسیر منحصر به فرد در درخت وجود دارد که از ریشه آغاز و در برگ به پایان می‌رسد. شکل ۱-۲ نمونه‌ای از این درخت را برای رشته TATAT نشان می‌دهد.



شکل ۱-۲. درخت پسوندی برای رشته TATAT

این درخت را می‌توان در زمان خطی نسبت به طول رشته ایجاد کرد. پس از ساخت درخت، درخواست‌هایی نظیر تطبیق دقیق رشته، یافتن تعداد تکرارها، یافتن طولانی‌ترین زیررشته و ... به راحتی قابل اجرا می‌باشد.

تطبیق دقیق رشته<sup>۱</sup>، یافتن محل‌هایی است که الگوی P در رشته رخ داده است. اگر این در محل i از رشته S رخ دهد، در اینصورت P پیشوندی از پسوند iام S می‌باشد. در حقیقت P با بخشی از یک مسیر در درخت پسوندی، مطابقت دارد. به کمک درخت پسوندی، می‌توان این مسئله را به راحتی حل کرد. نمونه‌ای از این مسئله برای  $S = TATAT$  و  $P = ATA$  در شکل ۱-۳ نشان داده شده است.

یافتن تعداد تکرارها، شامل تکرارهای دقیق<sup>۲</sup>، تکرار با K عدم تطابق<sup>۳</sup> و تکرار با K تفاوت<sup>۴</sup>

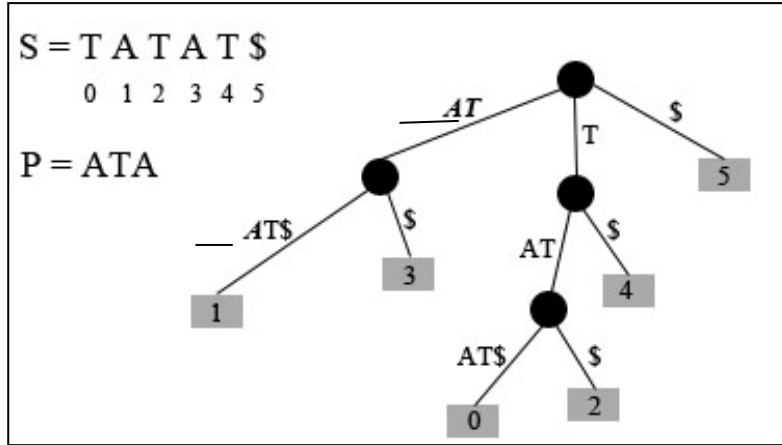
<sup>1</sup> Exact String Matching

<sup>2</sup> Exact repeat

<sup>3</sup> K-mismatch repeat

<sup>4</sup> K-differences repeat

می‌باشد. برای دو زیر رشته  $S[i_1..j_1]$  و  $S[i_2..j_2]$ ، این مسائل بصورت زیر تعریف می‌شوند:



شکل ۱-۳. یافتن الگوی ATA در رشته TATAT

اگر  $S[i_1..j_1] = S[i_2..j_2]$  باشد تکرار، دقیق است. اگر بین  $S[i_1..j_1]$  و  $S[i_2..j_2]$  دقیقاً  $k$  عدم تطابق وجود داشته باشد، تکرار با  $K$  عدم تطابق است. اگر  $k$  تفاوت (عدم تطابق، کمی و زیادی) بین  $S[i_1..j_1]$  و  $S[i_2..j_2]$  وجود داشته باشد، تکرار با  $K$  تفاوت است.

رشته  $S_3$  عبارت است از طولانی‌ترین زیررشته<sup>۱</sup>  $S_1$  (بقول  $n$ ) و  $S_2$  (به طول  $m$ )، که هم در  $S_1$  و هم در  $S_2$  رخ دهد. این مسئله به کمک ساخت درخت پسوندی تعمیم‌یافته<sup>۲</sup>، در زمان  $O(n+m)$  قابل حل است. با داشتن یک چنین درختی، کافی است طولانی‌ترین مسیر مشترک که از هر دو رشته دارای برگ است را در نظر بگیریم. شکل ۱-۴ نمونه‌ای از این کاربرد را برای دو رشته TATAT و AATAA نشان می‌دهد.

کاربردهای یاد شده، قدرت درخت‌های پسوندی را نشان می‌دهد، آنچنان که در (بارسکی<sup>۳</sup>، ۲۰۰۶) و (گاسفیلد<sup>۴</sup>، ۱۹۹۷) بیان شده است: «برای درک قدرت درخت پسوندی کافی است سعی کنیم تا این مسایل را بدون استفاده از درخت حل کنیم».

<sup>1</sup> longest common substring

<sup>2</sup> generalized suffix tree

<sup>3</sup> Barsky

<sup>4</sup> Gusfield