به نام خدا

موضوع مقاله: پیدا کردن کوتاه‌ترین ابر رشته‌ها در یک مجموعه از رشته‌ها

استاد درس: دکتر موسوی

اعضا:

ایمان تبریزیان، محمد مهدی سمیعی پاقلعه

فهرست

[مقدمه 3](#_Toc454767122)

[چکیده 3](#_Toc454767123)

[توضیح برخی نمادها و اصطلاحات 3](#_Toc454767124)

[اصطلاحاتی برای تقریب زدن الگوریتم‌های حریص تقریبی 4](#_Toc454767125)

[روش حل 4](#_Toc454767126)

[تلاش‌های انجام شده 5](#_Toc454767127)

[نقاط قوت 5](#_Toc454767128)

[نقاط ضعف 6](#_Toc454767129)

[مراجع 6](#_Toc454767130)

# مقدمه

مساله‌ی کوتاه‌ترین ابر دنباله‌ی مشترک یک مساله کلاسیک ترکیبیاتی بر روی رشته‌ها است، که کاربرد‌های فراوانی در حوزه‌های مختلف دارد. از این حوزه‌ها می‌توان به فشرده سازی داده‌ها، چسباندن تکه‌های DNA به یکدیگر و غیره اشاره کرد.

# چکیده

در ابتدای امر به توضیح مساله می‌پردازیم و سپس توضیحاتی درباره شیوه حل مساله و پیاده سازی آن ارائه می‌کنیم. مساله از این قرار است که ما یک مجموعه از رشته‌ها را در اختیار داریم و در صدد پیدا کردن کوتاه‌ترین ابررشته[[1]](#footnote-1) این مجموعه از رشته‌ها هستیم. ابررشته‌ی یک مجموعه از رشته‌ها، رشته‌ای است که در خود هر یک از آن رشته‌ها را داشته باشد، مثلا اگر باشد ابر رشته آن‌ها می‌تواند باشد. همین‌طور که مشاهده می‌کنیم ابررشته‌ی ارائه شده توسط ما دارای کوتاه‌ترین طول نمی‌باشد. در این مساله ما به دنبال یافتن کوتاه‌ترین طول هستیم.

در این مقاله ما با منظور کردن این تغییر در مساله اصلی که علاوه بر داشتن خود رشته می‌تواند معکوس آن را نیز داشته باشد که این نیز شرط ما ارضا می‌کند. یعنی به عنوان مثال در صورتی که یک ابررشته چه شامل خود رشته‌ی مجموعه‌ی ما چه اصل رشته‌ی ما باشد، از نظر ما این رشته در ابررشته است.

# توضیح برخی نمادها و اصطلاحات

***رخ دادن[[2]](#footnote-2) یک رشته در یک رشته‌ی دیگر :*** رشته‌ی t در رشته‌ی s رخ می‌دهد، اگر برای بعضی از v و z که از الفبای ما ساخته شده اند.

***پیشوند بودن[[3]](#footnote-3):*** اگر در تعریف رشته بالا v تهی باشد می‌گوییم که t پیشوندی برای رشته s است.

***پسوند بودن[[4]](#footnote-4):*** در تعریف رخ دادن اگر رشته‌ی z تهی باشد می‌گوییم که رشته‌ی z پسوندی برای رشته‌ی s است.

***معکوس یک رشته[[5]](#footnote-5):***  معکوس یک رشته پر واضح است که با خواندن رشته از راست به چپ رشته معکوس رشته داده شده ساخته می‌شود.

***Maximum Over lap:*** از نماد برای نشان دادن بیشترین هم‌پوشانی بین دو رشته u و v استفاده می‌کنیم. هم پوشانی هم به این صورت تعریف می‌شود که طول طولانی‌ترین رشته‌ای که پسوند آن u باشد و پیشوند آن v باشد.

# اصطلاحاتی برای تقریب زدن الگوریتم‌های حریص تقریبی

***Approximation ratio:*** طول رشته‌ی بدست آورده شده توسط الگوریتم حریص تقسیم بر طول بهترین رشته.

***Compression ratio***: برابر هست با که در آن n برابر است با مجموع طول رشته‌های موجود در S (همان مجموعه‌ای که قصد داریم ابررشته را برای آن بیابیم) است. و در آن طول بهترین ابررشته‌ی جواب و پارامتر دیگر طول رشته‌ی بدست آمده توسط الگوریتم حریص می‌باشد.

ما با استفاده از این اصطلاحات الگوریتم حریص ارائه شده در این مقاله را بررسی می‌کنیم. لازم به ذکر است که اثبات شده است که نرخ فشردگی همواره کمتر از ۰.۵ است. (به صورت آماری)

# روش حل

**ورودی**: یک مجموعه از رشته‌ها است.

**خروجی**: کوچکترین ابررشته که در آن فاکتور ها هم می‌توانند خود رشته یا معکسوشان باشد.

توضیح:

در ابتدا باید مجموعه‌ی داده شده را factor-free کنیم. factor-free کردن به این معناست که اگر رشته‌ای درون مجموعه‌ی ورودی، خودش یا معکوسش، فاکتور رشته‌ی دیگری بود آن رشته مفروض را که فاکتور آن رشته‌ی ثانویه است، حذف می‌کنیم. در مقاله به مقاله‌ی دیگری رجوع داده شده است که این عمل در زمان خطی امکان پذیر است.

مادامی که تعداد رشته های مجموعه s بزرگتر از ۱ است عمل زیر را تکرار می کنیم :

بین رشته های موجود درمجموعه s دو رشته ای که با هم بیشترین مقدار overlap را دارند را ‍‍پیدا می کنیم . این دو رشته را u,v و overlap(u,v) را w می نامیم .

w را به مجموعه s اضافه می کنیم.

u,v,ur,vr را از مجموعه s حذف میکنیم.

در نهایت تک رشته ای که در مجموعه s می ماند رشته جواب حاصل از الگوریتم حریص ماست.

# تلاش‌های انجام شده

دو دانشمند دیگر نیز تلاش کردند تا برای این مساله الگوریتم حریص تقریب بزنند که نرخ تقریب آن‌ها برابر تلاش کردند تا برای این مساله الگوریتم حریص تقریب بزنند که نرخ تقریب آن‌ها برابر است که باید به صورت عدد مخلوط خوانده شود. همچنین دانشمندان دیگری نیز نرخ تقریب‌های بالاتری را در گذشته ارائه کرده بودند.

مدل‌های دیگری از این مساله نیز ارائه شده است که در آن‌ها برای هر حرف مکمل آن نیز تعریف شده است و که در آن فاکتور بودن یک رشته برای یک رشته دیگر به معنی مکمل بودن آن رشته برای رشته‌ي دیگر است.

همچنین مدل اصلی آن که از سال‌های گذشته مطرح شده بود ورژن بدون مکمل و بدون معکوس بود.

# نقاط قوت

نقطه قوت این الگوریتم این است که نرخ فشرده‌سازی آن برابر ۰.۵ است که حداکثر مقدار فشرده سازی برای الگوریتم‌های حریصانه برای این مساله است.

# نقاط ضعف

نقطه ضعف این الگوریتم همانند دیگر الگوریتم‌های تقریبی برای مساله‌های NP-Complete این است که نرخ فشرده‌سازی آن حداکثر برابر ۰.۵ است.

در پیوست نیز پیاده‌سازی این کد ارسال می‌گردد.

# مراجع

[1] Esko Ukkonen, A linear-time algorithm for finding approximate shortest common superstrings, Algorithmica 5 (3) (1990) 313–323

[2] On the greedy algorithm for the Shortest Common Superstring problem with reversalsGabriele Fici, Tomasz Kociumaka, Jakub Radoszewski, Wojciech Rytter,Tomasz Walen

[3] https://en.wikipedia.org/wiki/Shortest\_common\_supersequence\_problem

1. Superstring [↑](#footnote-ref-1)
2. Occurrence [↑](#footnote-ref-2)
3. Prefix [↑](#footnote-ref-3)
4. Suffix [↑](#footnote-ref-4)
5. Reverse [↑](#footnote-ref-5)