تعرفنا في المقال السابق (https://www.syr-res.com/article/14175.html) على الطريقة الأولى لتسريع البحث عن كلمة معينة (تمثل السلسلة المطلوبة) ضمن نص كبير (يمثل الجينوم الكبير الذي نبحث فيه). نتعرف اليوم على الطريقة الثانية لتسريع البحث وهي باستخدام المؤتمتات.

ماذا نعني بالمؤتمتات؟
كما يتضح من الاسم، فإنّ المصطلح مرتبطٌ بعملية الأتمتة، أي جعلُ نظامَ إنتاجٍ لشيءٍ معين أوتوماتيكياً بدلاً من أن يكون يدوياً. المؤتمتات هي نماذجٌ لآلات تقوم بإجراء حسابات على مدخل معين، عبر الانتقال خلال مجموعة من الحالات أو الإعدادات. كلما وصلنا إلى حالة معينة، يتم تحديد الحالة التالية أو الإعداد التالي اعتماداً بشكلٍ جزئي على الحالة الحالية.

هناك العديد من الأنواع للمؤتمتات، سنتحدث عن النوع الذي سنستخدمه في خوارزمية اليوم وهو المؤتمتات المحدَّدة منتهية الحالات (DFA)، أي التي تملك عدداً محدداً من الحالات.

لنفترض أننا نملك مجموعةً كبيرةً من الكلمات ثلاثية الأحرف مثل bed، cat، hat وغيرها، ونريد أن نفصل الكلمات التي تحتوي على حرف a في الوسط.
يمكننا بناء مؤتمتة تقوم بعملية الفصل هذه، سوف نحتاج إلى حالةِ بداية وحالةِ نهاية ومجموعةٍ من الحالات الوسطية تشمل في مثالنا هذا الحالة الأولى والحالة الثانية.
ثم نبني تابع الانتقال من حالة البداية إلى الحالة التالية (الأولى)، بحيث ننتقل إليها عند وجود أي حرفٍ في الموقع الأول من الكلمة، لأننا لم نضع أيَّ شرطٍ على الموقع الاول.
وللانتقال من الحالة الأولى إلى الثانية يجب أن يتواجد حرف a في الموقع الثاني من الكلمة وهو الشرط الذي وضعناه، فإن لم يتحقق الشرط وتواجدَ حرفٌ آخر في الموقع الثاني من الكلمة، سوف نتوقف عند هذه الحالة ولن نعبر إلى الحالة التالية. أي أننا لن نصل إلى الحالة النهائية وبالتالي هذه الكلمة لم تحقق الشرط المطلوب.
أما إذا تحقق الشرط سوف نعبر إلى الحالة الثانية ومنها إلى النهائية التي لا تشترط أيَّ شرطٍ على الموقع الثالث من الكلمة، ويمكن الانتقال إليها لدى وجود أيِّ حرفٍ في الموقع الثالث من الكلمة. وبذلك نكون قد وصلنا إلى الحالة النهائية والكلمة قد حققت الشرط المطلوب.
لنقم بتمثيل ذلك بالشكل التوضيحي التالي:

[[[[img:28903]]]]

فإذا كانت الكلمات التي معنا هي : bed، hat، set، cat.
لنبدأ بـ bed: الحرف الأول هو b وسوف ننتقل من البداية إلى الحالة الأولى لأن الشرط تحقق وهو وجود أي حرف في الموقع الأول.
الحرف الثاني هو e وبذلك لن ننتقل إلى الحالة الثانية لأن الشرط لم يتحقق بوجود حرف a.
و بذلك توقفنا عند الحالة الثانية وليس عند الحالة النهائية، وفعلاً هذه الكلمة لا تحتوي على حرف a في الموقع الثاني وهو الشرط الذي وضعناه.

[[[[img:28904]]]]

لنأخذ كلمة hat: الحرف الأول هو h وسوف ننتقل من البداية إلى الحالة الأولى لأن الشرط تحقق وهو وجود أي حرف في الموقع الأول.
الحرف الثاني هو a وبالتالي يمكننا العبور إلى الحالة الثانية لأن الشرط تحقق وهو وجود الحرف a في الموقع الثاني.
الحرف الثالث هو t وسوف ننتقل إلى الحالة النهائية لأن الشرط تحقق وهو وجود أي حرف في الموقع الثالث.

[[[[img:28905]]]]

وهكذا نلاحظ أن هذه المؤتمتة تعبر عن الكلمات التي تحتوي حرف a في وسطها، ومجموع هذه الكلمات تمثل لغة تعبر عنها هذه المؤتمتة.

الآن كيف يمكن استخدام هذه المؤتمتات في تسريع عملية البحث؟
عندما نبحث عن كلمة معينة ضمن نص كبير، فإننا فعلياً نبحث هل يحتوي النص على هذه الكلمة؟ وهذا يشبه ما قمنا به للتو! فإن جعلنا الانتقالات بين الحالات تمثل حروف الكلمة التي نبحث عنها، ثم قمنا بفحص النص فإننا كلما وصلنا إلى الحالة النهائية يعني أننا وجدنا الكلمة المطلوبة.

لنفسر هذا الكلام بالمثال التالي:
إن كنا نبحث عن كلمة CAT ضمن النص التالي: ATCACAT
نقوم ببناء مؤتمتة تمثل الكلمة المطلوبة:

[[[[img:28906]]]]

الآن نقوم بتمرير النص عبر أجزاء كل منها بطول 3 أحرف:
نبدأ بـ ACT: الحرف الأول هو A لا يحقق شرط الانتقال من البداية إلى الحالة الأولى، وبهذا نتوقف عند البداية ولم نجد بالتالي الكلمة المطلوبة CAT.
نتابع مع TCA: الحرف الأول هو T لا يحقق شرط الانتقال من البداية إلى الحالة الأولى، وبهذا نتوقف عند البداية ولم نجد بالتالي الكلمة المطلوبة CAT.
ثم CAC: الحرف الأول C يحقق شرط الانتقال إلى الحالة الأولى، ثم الحرف الثاني A يحقق شرط الانتقال إلى الحالة الثانية، لكن الحرف الثالث C لا يحقق شرط الانتقال إلى الحالة النهائية وبهذا نتوقف عند الحالة الثانية ولم نجد بالتالي الكلمة المطلوبة.
ثم ACA: الحرف الأول هو A لا يحقق شرط الانتقال من البداية إلى الحالة الأولى، وبهذا نتوقف عند البداية ولم نجد بالتالي الكلمة المطلوبة CAT.
ثم CAT: الحرف الأول C يحقق شرط الانتقال إلى الحالة الأولى، ثم الحرف الثاني A يحقق شرط الانتقال إلى الحالة الثانية، والحرف الثالث T يحقق شرط الانتقال إلى الحالة النهائية وبهذا وجدنا الكلمة المطلوبة!

إذا أردنا تمثيل الحالات التي كانت نشطة أثناء عملية المقارنة السابقة سنحصل على التسلسل التالي، إذ أن كل حالة نشطة هي الحالة التي وصلنا إليها عبر الحرف السابق، ونلاحظ أن البداية دوماً نشطة:

[[[[img:28907]]]]

نلاحظ أنه يمكن لأكثر من حالة أن تكون نشطة في نفس الوقت، و لهذا يسمى هذا النوع من المؤتمتات بالمؤتمتات غير المحددة منتهية الحالات Automata Finite Deterministic-Non أو اختصاراً NFA.
إذاً يوجد كذلك مؤتمتات محددة منتهية الحالات Automata Finite Deterministic أو اختصاراً DFA وفيها تكون حالة واحدة فقط نشطة، هي تستخدم كذلك في تسريع البحث. فكيف يتم ذلك؟
بدايةً، كيف سيبدو شكل المؤتمتة السابقة لو قمنا بتمثيلها كـ DFA؟
يجب الالتزام بالتالي: من كلِّ حالةٍ يجب أن تخرج أسهم انتقالية تمثل كافة الحالات الممكنة.
أي سنحصل على الشكل التالي:

[[[[img:28908]]]]

نلاحظ أننا لا نعود دوماً إلى الحالة البدئية، و قد نعود للخلف وفقاً لسهم الانتقال، لكن كيف عرفنا مقدار العودة؟ أي عندما كنا في الحالة الثانية وكان الحرف التالي A كيف عرفنا أننا سنعود إلى البداية وليس إلى الحالة الأولى ولن نبقى كذلك في الحالة الثانية؟

لنفهمَ ذلك يمكننا العودة إلى شكلنا السابق وهو NFA والذي كان أكثر بساطة. لنتخيل الشكل الموافق من الـ NFA، فإنّ آخر حالة نشطة عندما ظهر الحرف الحالي في NFA هي الحالة التي نعود إليها لدى ظهور هذا الحرف في DFA!
مثال: عندما كنا في الحالة الثانية بعد ظهور الحرف A، ثم ظهر بعده الحرف C عندها كانت آخر حالة نشطة من اليمين هي الحالة البدئية، وعندها نعلم أن السهم في DFA سوف ينطلق من الحالة الثانية إلى البداية في حال ظهور الحرف C.
مثال2: عندما كنا في الحالة الأولى بعد ظهور الحرف C، لو ظهر بعده الحرف C كذلك عندها وفقاً للـ NFA ستكون كلا الحالتين البدئية والأولى نشطتين، و بما أن آخر حالة نشطة من اليمين هي الأولى فإذاً نعلم أن السهم في DFA سوف ينطلق من الحالة الأولى إلى الحالة الأولى ذاتها لدى ظهور C.
-هذه الخوارزمية بحساب الانتقال في DFA بالاعتماد على الـ NFA الموافق تدعى خوارزمية Morris-Knuth Pratt.

تعرفنا اليوم على إحدى طرق استخدام المؤتمتات في تسريع البحث. هناك العديد من الطرق الأخرى من استخدامها لتحديد إن كانت الكلمة نصاً جزئياً من الجزء الحالي أم تقع في بدايته.
نتابع في المقال التالي الطريقة التالية وهي باستخدام البُنى الشجرية للمعطيات.

المصادر:
Algorithms for Sequence Analysis Lecture Notes- Saarland University
Flexible Pattern Matching in Strings: Practical On-Line Search Algorithms for Texts and Biological Sequences، Navarro،G. and Raffinot،M.
http://www.eecs.wsu.edu/~ananth/CptS317/Lectures/IntroToAutomataTheory.pdf
-https://cs.stanford.edu/people/eroberts/courses/soco/projects/2004-05/automata theory/basics.html
http://www.eecs.wsu.edu/~ananth/CptS317/Lectures/FiniteAutomata.pdf

المساهمون في المقال :


إعداد: Dania S. Humaidan


تدقيق علمي: Bassel Zeno


تدقيق لغوي: Wasim Dimashky


تصميم الصورة: Ramy Ali


صوت: Ghandi Safar Saado


نشر: Sandra Sukarieh


تعديل: Sandra Sukarieh