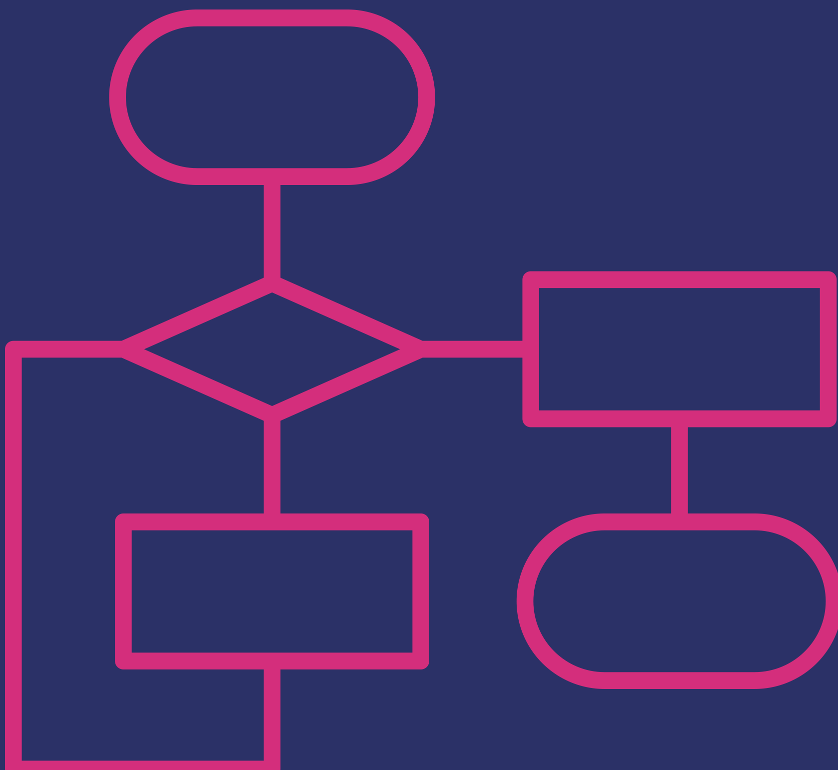
بانوس لوريداس

الخوارزميات

ترجمة إبراهيم سند أحمد



سلسلة المعارف الأساسية

الخوارزميات

تأليف
بانوس لوريداس

ترجمة
إبراهيم سند أحمد

مراجعة
شيماء طه الريدي

هنداوي

الخوارزميات

بانوس لوريداس

Algorithms

Panos Louridas

الناشر مؤسسة هنداوي
المشهرة برقم ١٠٥٨٥٩٧٠ بتاريخ ٢٦ / ١ / ٢٠١٧

يورك هاوس، شييت ستريت، وندسور، SL4 1DD، المملكة المتحدة
تليفون: ٨٣٢٥٢٢ ١٧٥٣ (٠) ٤٤ +
البريد الإلكتروني: hindawi@hindawi.org
الموقع الإلكتروني: https://www.hindawi.org

تصميم الغلاف: يوسف غازي

الترقيم الدولي: ٢ ٢٧٤١ ٥٢٧٣ ١ ٩٧٨

صدر الكتاب الأصلي باللغة الإنجليزية عام ٢٠٢٠.
صدرت هذه الترجمة عن مؤسسة هنداوي عام ٢٠٢٢.

المحتويات

تمهيد السلسلة

تُقدِّم «سلسلة المعارف الأساسية» التي تَنشرها مؤسسة «إم آي تي بريس» كُتبًا موجزة بلغةٍ جَزلة سهلة الفهم، وشكلٍ أنيق، وحجمٍ صغير يُلائم الجيب، تُناقِش الموضوعات التي تُثير الاهتمام في الوقت الحالي. ولمَّا كانت كُتب هذه السلسلة من تأليف مفكرين بارزين، فإنها تُقدِّم آراء الخبراء بشأن موضوعاتٍ تتنوَّع بين المجالات الثقافية والتاريخية، إضافةً إلى العِلمية والتقنية.

في ظلِّ ما يَشيع في هذا العصر من إشباعٍ لَحظي للمعلومات، أضحى لدى الجميع القدرةُ على الوصول إلى الآراء والأفكار والشروح السطحية بسرعةٍ وسهولة، وأصبح من الصعوبة بمكانٍ أن يَحظى المرءُ بالمعرفة الأساسية التي تُيسِّر فَهمًا صادقًا للعالَم؛ وما تفعله كُتب هذه السلسلة هو أنها تُحقِّق ذلك الغرض. وكل كتابٍ من هذه الكُتب المختصَرة يُقدِّم للقارئ وسيلةً مُيسَّرة للوصول إلى الأفكار المعقَّدة، من خلال تبسيط المواد المُتخصِّصة لغير المُختصِّين، وشرْح الموضوعات المهمة بأبسطِ طريقةٍ ممكنة.

بروس تيدور
أستاذ الهندسة البيولوجية وعلوم الكمبيوتر
«معهد ماساتشوستس للتكنولوجيا»

هذا العالَم غير قابل للتفسير، ولكن هذا لا يعني أنه غير قابل للفهم، ما دمتَ تعرف تلك القاعدة البسيطة، من أن لا شيء مما يُعبِّر عنه من خلال مخلوقاته وأرواحه العديدة يتبعه علامات استفهام، بل علامات تعجُّب فحسب.

كارل أوفه كنوسجارد، «الصيف»

مقدمة

أعرف صبيين مراهقين بحوزتهما معرفةٌ تفوق أيَّ عالم أو فيلسوف أو باحث في العصور الماضية. إنهما ولداي. كلا، لستُ أبًا خرِفًا يزهو بمواهب أبنائه غير العادية. ولكنَّ هذين الصبيين يحملان في جيوبهما أجهزةً توصلهما بمستودعٍ هائل لم يُخلق مثله من المعلومات. لا يوجد سؤال واقعي تُعجِزهما الإجابة عنه، حتى إنهما باتا الآن يتقنان فنَّ معرفة أماكن البحث على شبكة الإنترنت. بإمكانهما الترجمة إلى اللغات الأجنبية ومنها، دون الاضطرار إلى تصفُّح المعاجم الضخمة، التي لا نزال نحتفظ بها في المنزل كي يعرف أطفالنا كيف كانت تجري الأمور قبل بضع سنين فقط. وتصل إليهم الأخبار من أي مكان في لحظة. كذلك بإمكانهم التواصلُ مع أقرانهم من أي مكان في العالم مهما كان قبل أن نعرف مَن هم. ويستطيعون التخطيط لمواعيد الخروج بتفصيل رائع. لكن للأسف، يهدرون وقتَهم في ممارسة الألعاب بإفراط ومتابعة الموضوعات الرائجة التي تتغيَّر بسرعة شديدة لدرجة أني لا أعرف مكمن أهميتها.

كلُّ ما ذُكر آنفًا أصبح سهلًا بفضل التطورات الهائلة في التكنولوجيا الرقمية. فقد أصبح في جيوبنا الآن إمكانيات حوسبة أكبر مما استُخدم لنقل الإنسان إلى القمر. وكما يتبيَّن من المثال الذي ضربته بهذين الصبيين، كانت التغيُّرات التي طرأت على حياتنا هائلة؛ تتراوح التنبؤات للمستقبل بين الواقع المثالي، حيث لن يحتاج الناس حقًّا إلى العمل، والواقع المرير، حيث لن يهنأ بالحياة سوى القلة من أصحاب الامتيازات، بينما سيقبع البقية في سبات الكسل بلا قيمةٍ تُذكر. لحسن الحظ أننا قادرون على تشكيل هذا المستقبل، والعامل المهم في قدرتنا على القيام بذلك هو مدى إلمامنا بالتقنيات التي تقف خلف الإنجازات والتغييرات التي أمامنا. نحن نعيش في أفضل فترة من التاريخ البشري على الرغم من أننا

قد لا نرى ذلك في خضم صخب الحياة اليومية. فنحن نتمتع بصحةٍ أفضل مما كنا عليه في أي وقتٍ مضى، ويُتوقع أن نحظى بأعمار أطول من أي جيل سبقَنا على الإطلاق. وعلى الرغم من الظلم الناجم عن عدم المساواة الفجة، فإن أعدادًا مهولة من البشر تخلصوا من أغلال الفقر. لم يسبق أن اقترب بعضنا من بعض — سواء بالمعنى الافتراضي أو الحرفي — على هذا النحو من قبل. ربما نستهجن الطابع التجاري للسياحة العالمية الجماعية، ولكن انخفاض تكاليف السفر يتيح لنا معايشة ثقافات مختلفة، وزيارة أماكن لم يكن يسعنا قبل ذلك إلا الإعجاب بصورها المتناثرة في الكتب على طاولة القهوة. كل هذا التقدُّم يمكن أن يستمر، وينبغي أن يستمر.

غير أن المشاركة في هذا التقدُّم تقتضي عدم الاكتفاء باستخدام التكنولوجيا الرقمية. لا بد أن نكون قادرين على فهمها. أولًا: لأنها، من الناحية العملية البحتة، توفِّر فرص توظيف ممتازة. ثانيًا: حتى لو لم نكن مهتمين بالعمل في مجال التكنولوجيا، فلا بد أن نعرف مبادئها الأساسية كي نقدِّر إمكانياتها ونشكِّل دورنا في استخدامها. تكمن إمكانيات التكنولوجيا الرقمية في مكوِّنات أجهزتها — المكونات المادية التي تتركَّب منها أجهزة الكمبيوتر والخدمات الرقمية — بقدرِ ما تكمن في برمجياتها؛ أي البرامج التي تشغِّلها. والعنصر الأساسي في تلك البرامج هو الخوارزميات التي تطبِّقها؛ أي مجموعة التعليمات التي تصف طريقة حل مسائل بعينها (وإن لم يوضح هذا التعريف معنى الخوارزميات، فلا تقلق لأننا سنتناوله بالتفصيل على مدى ما تبقى من الكتاب). من دون الخوارزميات، لم نكن لنرجو فائدةً من أجهزة الكمبيوتر، وما وُجد أيٌّ من أنماط التكنولوجيا الحديثة.

ما يجب أن نعرفه هو التغييرات التي طرأت على مدى مدة من الزمن. على مدى الجزء الأكبر من تاريخ البشرية، لم يكن التعليم المدرسي يعتبر ضروريًّا على الإطلاق. فكان معظم الناس أميين، وإن تعلموا شيئًا، كان يتمثَّل في إتقان مهارة حِرفية أو تعلُّم الكتاب المقدَّس. في بداية القرن التاسع عشر، كان أكثر من ٨٠ بالمائة من سكان العالم لا يذهبون إلى المدارس البتة؛ أما الآن فباتت الغالبية العظمى تتلقى التعليم بالمدارس سنواتٍ عديدة ويُتوقع أن تبلغ نسبة غير الملتحقين بالمدارس في العالم صفرًا في نهاية القرن. كذلك زاد عدد السنوات التي نقضيها في التعليم. ففي عام ١٩٤٠، كانت نسبة حاملي الشهادات الجامعية من الأمريكيين أقل من ٥ بالمائة، ولكن بحلول عام ٢٠١٥، أصبح ثلثهم تقريبًا يحمل شهادة جامعية.[1]

تكمن التكنولوجيا الرقمية في مكوِّنات أجهزتها — المكونات المادية التي تتركَّب منها أجهزة الكمبيوتر والخدمات الرقمية — بقدر ما تكمن في برمجياتها؛ أي البرامج التي تشغلها. والعنصر الأساسي في تلك البرامج هو الخوارزميات التي تطبقها.

في القرن التاسع عشر، لم تكن المدارس تُدرِّس علم الأحياء الجزيئي؛ لأن أحدًا لم يكن يعرف شيئًا عنه؛ فالحمض النووي لم يُكتشف إلا في القرن العشرين. أما الآن، فبات يشكل جزءًا مما يُعد أساس القاعدة المعرفية للشخص المتعلم. بالمثل، على الرغم من اكتشاف الخوارزميات في العصور القديمة، فإن قلة من الناس انزعجوا من تعلُّمها حتى ظهور أجهزة الكمبيوتر الحديثة. يترسَّخ في اعتقاد المؤلِّف أننا قد وصلنا إلى مرحلةٍ أصبحت فيها الخوارزميات في قلب العلوم التي نعتبرها المعارف الأساسية. وما لم نعرف ما هي الخوارزميات والآلية التي تعمل بها، لن نفهم ما يمكن أن تفعله وكيف يمكن أن تؤثِّر فينا، وما المتوقَّع من تعلُّمها وما حدودها وما الذي يتطلَّبه العمل بها. وفي مجتمعٍ تتزايد فيه وتيرة العمل بفضل الخوارزميات، فحري بنا، كمواطنين على قدرٍ من الثقافة والتعليم، أن نكون على دراية بها.

من الممكن أيضًا أن نستفيد من تعلُّم الخوارزميات بصورةٍ أخرى. إذا كان تعلُّم الرياضيات يعرِّفنا على طريقة للاستدلال المنطقي الدقيق؛ فالإلمام بالخوارزميات يعرِّفنا على طريقةٍ جديدة للتفكير الخوارزمي؛ وهي طريقة لحل المسائل بطريقةٍ عملية، بحيث يمكن للتطبيقات الفعَّالة للخوارزميات كبرامج أن تعمل بسرعة على أجهزة الكمبيوتر. فالتركيز على تصميم عملياتٍ فعَّالة وعملية يمكن أن يكون أداةً عقلية مفيدة، حتى لو لم نكن مبرمجين احترافيين.

الإلمام بالخوارزميات يعرِّفنا على طريقةٍ جديدة للتفكير الخوارزمي؛ وهي طريقةٌ لحل المسائل بطريقةٍ عملية، بحيث يمكن للتطبيقات الفعَّالة للخوارزميات كبرامج أن تعمل بسرعة على أجهزة الكمبيوتر.

يهدف هذا الكتاب إلى تقديم الخوارزميات لغير المتخصصين بطريقةٍ تجعل القارئ يفهم طريقة عمل الخوارزميات في الواقع. ليس الغرض وصف تأثيرات الخوارزميات في حياتنا؛ فهناك كتب أخرى تبلى بلاء حسنًا في توضيح مدى التغيُّر الذي قد يطرأ على

الظروف والأوضاع البشرية بفضل تحسين معالجة البيانات الضخمة والذكاء الاصطناعي ودمج أجهزة الحوسبة في نسيج الحياة اليومية. في هذا الكتاب، لن نهتم «بما» قد يحدث، بل «بالطريقة» التي يمكن أن يحدث بها. وحتى نحقِّق ذلك، سنعرض خوارزميات حقيقية ولن نوضح ما تفعله فحسب، بل سنوضح طريقة تطبيقها فعليًّا أيضًا. فبدلًا من مجرد التلميحات، سنقدِّم شروحات مفصَّلة.

بالنسبة إلى سؤال «ما هي الخوارزميات؟» الإجابة غاية في البساطة. إنها طرقٌ خاصة لحل المسائل. يمكن سرد تلك الطرق في حل المسائل في خطواتٍ سهلة بحيث يمكن لأجهزة الكمبيوتر أن تنفِّذها بسرعة وكفاءة مذهلتين. ومع ذلك فلا يوجد شيء سحري في هذه الحلول. وحقيقة أنها تتكوَّن من خطوات أساسية بسيطة تعني أنه لا يوجد ما يدعو للاعتقاد بأنها تتجاوز قدرة غالبية الناس على الفهم.

في الحقيقة، لا يدَّعي الكتاب معرفته بمادة تفوق ما يدرس عادة في المدارس الثانوية. هناك بالفعل بعض المسائل الرياضية تظهر في بعض صفحات الكتاب؛ لأنه لا يمكن التحدُّث بجدية عن الخوارزميات من دون استخدام «بعض» الرموز. كذلك يتناول الكتاب أيَّ مفاهيم شائعة الاستخدام في الخوارزميات، ولكنها قد لا تكون شائعةً خارج مجال علوم الكمبيوتر.

كتب عالم الفيزياء الراحل ستيفن هوكينج في مقدمة أفضل كتبه مبيعًا «تاريخ موجز للزمن»، المنشور عام ١٩٨٨ يقول: «قال لي أحدهم إن كل معادلة أدرجتها في الكتاب ستهوي بالمبيعات إلى النصف.» تبدو هذه إشارة غير مبشِّرة للكتاب الذي بين يدينا؛ نظرًا لظهور مسائل رياضية أكثر من مرة على مداه. لكني قررت المتابعة لسببين. السبب الأول: لما كان مستوى إجادة الرياضيات المطلوب لفهم فيزياء هوكينج يحتاج إلى شخص في المرحلة الجامعية أو ما بعدها كي تُفهم؛ فالمسائل الرياضية المقدَّمة هنا أسهل بكثير. السبب الثاني: أن الكتاب لا يهدف إلى التعريف بالخوارزميات فحسب، بل يهدف إلى توضيح آلية عملها أيضًا؛ ومِن ثَم ينبغي للقارئ أن يلمَّ ببعض المصطلحات التي نستخدمها عند مناقشة الخوارزميات. وتلك المصطلحات تحتوي على بعض الرياضيات. فالرموز ليست امتيازًا مقصورًا على طبقة التقنيين فحسب، والإلمام بها سيعين على تبديد أي غموض يحيط بالموضوع؛ ففي النهاية، سنرى أن الأمر يعتمد اعتمادًا كبيرًا على القدرة على التحدُّث عن الأشياء من منظور كمي دقيق.

لا يمكن تغطية موضوع الخوارزميات بالكامل من خلال كتابٍ كهذا، ولكن يمكن أن نقدِّم للقارئ نظرة عامة ونعرِّفه على طريقة التفكير الخوارزمي. يضع الفصل الأول

حجر الأساس من خلال تعريف الخوارزميات وكيف يمكن قياس كفاءتها. في البداية، يمكننا القول إن الخوارزمية عبارة عن تسلسل محدود من الخطوات يمكن أن ننفذها باستخدام ورقة وقلم، وهذا التعريف البسيط ليس بعيدًا عن الواقع. يبدأ الفصل الأول من تلك النقطة، وفي الوقت نفسه يوضِّح أيضًا العلاقة بين الخوارزميات والرياضيات. من الفوارق الأساسية بين المجالين الجانبُ العملي؛ ففي الخوارزميات، نهتم بالطرق العملية في حل المسائل. وهذا يعني أنه ينبغي أن نتمكَّن من قياس مدى كفاءة الخوارزميات وفاعليتها العملية. وسنرى أنه يمكن صياغة تلك الأسئلة بعناية من خلال فكرة التعقيد الحسابي؛ وهذا من شأنه أن يشكِّل ملامح مناقشة الخوارزميات في بقية الكتاب.

تتناول الفصول الثلاثة التالية ثلاثة من أهم المجالات التطبيقية للخوارزميات. يتحدَّث الفصل الثاني عن الخوارزميات التي تتعامل مع حل المسائل المرتبطة بشبكات الأشياء، التي يُطلق عليها التمثيلات البيانية. قد تتضمن تلك المسائل البحث عن طريقٍ في شبكة طرق أو سلسلة علاقات تربطك بشخصٍ ما في شبكة اجتماعية، وتتضمَّن أيضًا مسائل في مجالات أخرى تربط بينها علاقات، ولكن ليست واضحة، مثل: تسلسل الحمض النووي وجدولة المسابقات؛ تلك الموضوعات ستوضح إمكانية إيجاد حلٍّ ناجز لمسائل مختلفة باستخدام الأدوات نفسها.

يتناول الفصلان الثالث والرابع طريقةَ البحث عن العناصر وترتيبها. قد تبدو تلك العمليات عادية، ولكنها من أهم التطبيقات في أجهزة الكمبيوتر. تستغرق أجهزة الكمبيوتر وقتًا طويلًا في التصنيف والبحث، ولكننا غافلون إلى حدٍّ كبير عن تلك الحقيقة لأنها جزء أصيل غير مرئي من العديد من التطبيقات. يقدِّم لنا موضوعا التصنيف والبحث أيضًا لمحةً عن وجهٍ مهمٍّ للخوارزميات. في العديد من المسائل، نعرف أكثر من خوارزمية لحلِّها. ونحن ننتقي من بين الخوارزميات المتاحة بناءً على سماتها الخاصة المميزة لها؛ فبعض الخوارزميات تناسب مسائلَ معينةً أكثر من غيرها. ومن هذا المنطلق، ينبغي التعرُّف على كيفية تعامُل الخوارزميات المختلفة — ذات السمات المختلفة — مع حلول المسألة الواحدة.

يعرِض الفصلان التاليان تطبيقاتٍ مهمة للخوارزميات على نطاقٍ واسع. ويعاود الفصل الخامس الحديث عن التمثيلات البيانية ويشرح خوارزمية بيج رانك التي يمكن استخدامها في ترتيب صفحات الويب حسب أهميتها. وقد كانت بيج رانك الخوارزمية التي تستخدمها شركة «جوجل» وقت إنشائها. وقد لعب نجاح الخوارزمية في ترتيب صفحات الويب في نتائج البحث دورًا بالغ الأهمية في النجاح المذهل الذي حقَّقته «جوجل» كشركة.

ولحسن الحظ، ليس من الصعب فهم آلية عمل خوارزمية بيج رانك. وستوفِّر لنا الفرصة لمعرفة كيف يمكن لخوارزميةٍ ما حلَّ مسألةٍ تبدو للوهلة الأولى مستعصية الحل عن طريق الكمبيوتر: كيف نقيس درجة الأهمية؟

يتناول الفصل السادس واحدًا من أهم المجالات النشطة في علوم الكمبيوتر، ألا وهو الشبكات العصبية والتعلُّم العميق. والتطبيقات الناجحة للشبكات العصبية أمرٌ مُثار في وسائل الإعلام العامة. تثير القصص اهتمامنا من خلال الحديث عن الأنظمة التي تنفِّذ مهامَّ، مثل تحليل الصور أو الترجمة الآلية أو التشخيص الطبي. وسنبدأ من نقطةٍ بسيطة وهي الخلايا العصبية الفردية التي تبني شبكاتٍ عصبية أكبر وأكبر قادرةً على تنفيذ مهام معقَّدة أكثر وأكثر. سنرى أن جميع تلك الشبكات تعمل بناء على بعض المبادئ الجوهرية. تستمد الشبكات كفاءتها من الترابط بين العديد من المكوِّنات البسيطة وتطبيق خوارزمية تتيح لتلك الشبكات العصبية إمكانية التعلُّم.

بعد توضيحِ ما يمكن أن تفعله الخوارزمية، تستعرض الخاتمة حدود عمليات الحوسبة. نعلم أن أجهزة الكمبيوتر حقَّقت نجاحاتٍ باهرة ونتوقَّع منها المزيد في المستقبل، ولكن هل توجد أشياء لا يمكن لأجهزة الكمبيوتر إنجازها؟ سيتيح الحديث عن حدود عمليات الحوسبة تقديمَ شرحٍ أدقَّ لطبيعة الخوارزميات والحوسبة. كنا قد قلنا إنه يمكن وصف الخوارزمية بأنها سلسلة محدودة من الخطوات يمكن تنفيذها باستخدام ورقة وقلم، ولكن أي نوع من الخطوات تلك التي يمكن أن تتضمَّنها الخوارزمية؟ وما مدى ارتباط تشبيه الورقة والقلم بماهية الخوارزميات في الحقيقة؟

شكر وتقدير

بداية أودُّ أن أتقدَّم بالشكر إلى كلٍّ من ماري لوفكين لي بمطبعة معهد ماساتشوستس للتكنولوجيا، على طرحها فكرةَ هذا الكتاب، وستيفاني كوهين على تحفيزي بلطف أثناء تأليف الكتاب، وسيندي ميلستين على تحريرها الدقيق، وفيرجينيا كروسمان على اهتمامها الرائع بالتفاصيل والعناية بكل شيء. يجب إدراج كتاب عن الخوارزميات ضمن سلسلة «المعارف الأساسية»، وأنا فخور بأنني مَن دوَّن ذلك الكتاب.

كذلك أقدِّم شكري إلى ديوميديس سبينليس لما أبداه من تعليقاتٍ حول بعض أجزاء الكتاب، وأتقدَّم بشكرٍ خاصٍّ إلى كونستانتينوس مارنياكوس الذي اطلع على نص الكتاب وكشفَ أخطاءً قاتلة، وتكرَّم بعرض اقتراحات للتنقيح.

أخيرًا، أودُّ التعبير عن امتناني إلى ولديَّ؛ أدريانوس وإكتور، اللذين سيتحدَّد مسار حياتهما إلى حدٍّ ما بموضوع هذا الكتاب، وكذلك إلى أمهما إليني؛ فبفضلهم تمكَّنت من إنجاز هذا الكتاب.

الفصل الأول

ما هي الخوارزمية؟

عصر الخوارزميات

نحب أن نطلق مسمياتٍ على الفترات الزمنية؛ ربما لأن إلحاقَ اسمٍ بفترة زمنيةٍ ما يتيح لنا فهم متغيراتها ومجرياتها. ولذلك بدأنا الحديث عن الوقت الحاضر بوصفه فجرًا لـ «عصرٍ خوارزمي» جديد ستكون فيه السيطرة للخوارزميات وستحكم المزيد والمزيد من مناحي الحياة. ومن المثير للاهتمام أننا لم نَعُد نتحدَّث عن «عصر الكمبيوتر» أو «عصر الإنترنت». فقد بتنا نتعامل مع ذلك كمسلمات. أما عندما نضيف الخوارزميات، فإننا نبدأ في التلميح إلى أن شيئًا مختلفًا من المنظور النوعي بدأ يلوح في الأفق. يقول كريستوفر ليدون، الصحفي السابق في جريدة «نيويورك تايمز» ومقدِّم برنامج «راديو أوبن سورس»: «انظروا إلى قوة الخوارزميات، جزء من شفرة كمبيوتر آتية كي تمثِّل سلطةً عليا في عصرنا العلماني، أقرب إلى إله.» والواقع أن الخوارزميات تعتبر فعلًا شكلًا من أشكال السلطة العليا حين تُستخدم لتنظيم الحملات السياسية، وتتبُّع آثارنا في عالم الإنترنت، أو تعقُّب عمليات التسوُّق واستهدافنا بالإعلانات، أو اقتراح شركاء للمواعدة، أو متابعة حالتنا الصحية.[1]

ثمَّة هالة من الغموض تطوِّق كل ذلك، الأمر الذي ربما يداعب خُيلاء أنصار الخوارزميات. إن وصف «مبرمج» أو «عالم كمبيوتر» يصف المرء بأنه شخصية جديرة بالاحترام، وإن كان وصفًا تقنيًّا نوعًا ما. إلى أي مدًى تفضِّل أن تكون فردًا في جماعةٍ توشك أن تغيِّر كل شيء تقريبًا في حياتنا؟

لا شكَّ أن ثمَّة مغزًى من وصف الخوارزميات بأنها أقرب إلى إله. ففي أغلب الأحيان تكون في منأًى عن المساءلة مثل الآلهة؛ فالأشياء تحدث ليس بسبب قدرة البشر، ولكن لأن مَن قرَّر حدوثها خوارزمية، والخوارزمية فوق مستوى المساءلة. والمجالات التي يمكن أن

تتفوَّق فيها الأجهزة التي تعمل بالخوارزميات على الأداء البشري في تزايد؛ لدرجة أن نطاق تفوُّق البشر يبدو في تناقصٍ يومًا بعد يوم، ويعتقد البعض أن اليوم الذي سنرى فيه قدرة أجهزة الكمبيوتر على التفوق على البشر في مجالات المعرفة كافة بات قريبًا.

لكن ثمَّة جانبًا أيضًا لا تتشابه فيه الخوارزميات مع الآلهة على الرغم من أننا لا نراه في كثير من الأحيان. فالخوارزميات لا تظهر نتائجها بالبوح الصريح. نحن نعلم تمامَ العلم القواعدَ التي تتبعها ونوعية الخطوات التي تتخذها. ومهما كانت روعة النتائج، يمكن إرجاعها دومًا إلى بعض العمليات البسيطة. وقد يتفاجأ المستجدون حديثو العهد بالخوارزميات من مدى سهولة تلك العمليات. وهذا لا يقلل من شأن الخوارزميات؛ فمعرفة الطريقة التي يسير بها شيءٌ ما يمكن أن تزيل جزءًا من غموضه. وفي الوقت نفسه، يتيح لنا فهمُ طريقة عمل شيء تقديرَ روعة تصميمه حتى لو لم يَعُد غامضًا.

يقوم هذا الكتاب على فرضية أن الخوارزميات ليست غامضة في الواقع. إنها أدوات تتيح لنا حُسن إنجاز أشياء معينة؛ إنها أنواع محدَّدة من الأدوات الهدفُ منها أن تتيح لنا حلَّ المسائل. وهي بهذا المعنى تعتبر أدواتٍ معرفية؛ ولكنها بذاتها ليست الأدوات المعرفية الوحيدة. فالأعداد والعمليات الحسابية أيضًا من الأدوات المعرفية. وقد استغرق الأمر آلاف السنين حتى طوَّر الإنسان نظامًا للأعداد يسهُل على الأطفال تعلُّمه في المدارس بحيث يمكنهم إجراء العمليات الحسابية التي كان ليستحيل إجراؤها من دون ذلك النظام. لقد صارت إجادة مبادئ علم الحساب من المسلَّمات الآن، ولكن قبل بضعة أجيال، لم يكن يعرفه سوى قلة قليلة من البشر.

بالمثل، ينبغي ألَّا تكون الخوارزميات امتيازًا لقلةٍ قليلة من الصفوة؛ فنظرًا لكونها أدواتٍ معرفية، فإن بإمكان البشر بشتَّى أطيافهم فهْمَها، وليس فقط محترفو الكمبيوتر. إضافة إلى ذلك، «ينبغي» أن يفهم مزيدٌ من الناس الخوارزميات؛ لأن ذلك سيتيح لنا وضعها في منظورها الصحيح؛ أي معرفة ما تقوم به الخوارزميات، وكيف تقوم به، وما الذي يمكن أن نتوقَّعه منها فعليًّا.

إن ما نرمي إليه في هذا الكتاب هو اكتساب معرفةٍ أساسية بالخوارزميات بحيث يمكننا أن نشارك بجزء هادف في الأحاديث عن عصر الخوارزميات. هذا العصر ليس مفروضًا علينا، بل هو من صنْع أيدينا، وقائم على أدواتٍ نحن مَن اخترعناها. ودراسة هذه الأدوات هي موضوع هذا الكتاب. الخوارزميات أدوات رائعة، ومعرفة لمحة عن طريقة تركيبها وعملها يمكن أن يساعدنا في تعزيز طريقة تفكيرنا.

سنبدأ بدحض فكرةٍ مزعجة، وهي أن الخوارزميات تتعلَّق بأجهزة الكمبيوتر. وهذا منطقي — كما سنرى — كقول إن الأعداد مرتبطة بالآلات الحاسبة.

طريقة لإنجاز المهام

أحجية من أحاجي الورقة والقلم، وموسيقى، ومجموعة متنوعة من الأعداد، ومسرِّعات نيوترونات في فيزياء الجسيمات؛ سنرى أن العامل المشترك بين هذه الأشياء جميعًا هو الخوارزمية نفسها، المطبَّقة في تلك المجالات على اختلافها، ولكنها قائمة على المبادئ الأساسية نفسها. كيف يمكن هذا؟

كلمة «خوارزمية» في ذاتها لا توضِّح معناها. الاسم مشتق من اسم محمد بن موسى الخوارزمي (من عام ٧٨٠ إلى عام ٨٥٠ تقريبًا)، وهو عالِم فارسي في الرياضيات والفلك والجغرافيا. تعدَّدت إسهامات الخوارزمي وانتشرت على نطاق واسع. فمصطلح «الجبر» مشتقٌّ من العنوان العربي لأكثرِ كتبه تأثيرًا وهو كتاب «المختصر في حساب الجبر والمقابلة». كان ثاني كتبه تأثيرًا كتاب «عن الحساب بالأرقام الهندية»؛ إذ تناول العمليات الحسابية وقدَّم نظام الأعداد العربية–الهندية إلى العالَم الغربي من خلال ترجمته إلى اللاتينية. أُدخل اسم الخوارزمي إلى اللغة اللاتينية وصار Algorismus والذي أصبح يشير إلى طريقة الحساب العددي باستخدام الأعداد العشرية. تأثَّر المصطلح اللاتيني Algorismus بالكلمة اليونانية arithmos وتعني «العدد» (مثل كلمة arithmetic وتعني علم الحساب)، ومِن ثَم أصبحت algorithm بمعنى خوارزمية، وظلت تشير إلى العمليات الحسابية العشرية، قبل أن تكتسب معناها الحديث في القرن التاسع عشر.

قد يميل المرء إلى الاعتقاد بأن الخوارزميات شيء نقوم به بواسطة أجهزة الكمبيوتر، ولكن هذا الاعتقاد خاطئ. ويُعزى خطؤه إلى أن الخوارزميات كانت موجودة قبل اختراع أجهزة الكمبيوتر بزمن طويل. فيعود تاريخ أول خوارزميات معروفة إلى الحضارة البابلية القديمة.[2] كذلك يُعزى الخطأ إلى أن الخوارزميات ليس لها علاقة بالمسائل المرتبطة بأجهزة الكمبيوتر. فالخوارزميات تدور حول فعل شيءٍ ما بطريقة محدَّدة وباتباع سلسلة خطوات معينة. هنا يكمن بعض الغموض. فقد تتساءل، ما نوع تلك الخطوات؟ وما تلك الطريقة المحددة؟ يمكننا تبديد هذا الغموض برمَّته وتقديم تعريف رياضي دقيق لمعنى الخوارزمية وما تقوم به — وهذا التعريف موجود بالفعل — ولكننا لسنا بحاجةٍ إلى كل هذا. قد تُسرُّ عندما تعرف أن الخوارزمية مجموعةُ خطوات يمكن أن تتَّبعها باستخدام ورقة

وقلم، ويمكن أن تطمئن إلى أن هذا الوصف الذي يبدو مبسطًا قريبٌ من التعريفات التي يستخدمها علماء الرياضيات وعلوم الكمبيوتر.

> قد يميل المرء إلى الاعتقاد بأن الخوارزميات شيء نقوم به بواسطة أجهزة الكمبيوتر، ولكن هذا الاعتقاد خاطئ. ويُعزى خطؤه إلى أن الخوارزميات كانت موجودة قبل اختراع أجهزة الكمبيوتر بزمن طويل.

ومِن ثَم يمكننا البدء في شرح الخوارزمية بمسألةٍ يمكن حلُّها باستخدام الكتابة فقط. لنفترض أن لدينا مجموعتين من الأشياء، ونريد توزيع المجموعة الأولى من الأشياء بين الأشياء في المجموعة الثانية بالتساوي قدْر الإمكان. سنستخدم علامة (×) للتعبير عن الأشياء في المجموعة الأولى وعلامة (•) للتعبير عن الأشياء في المجموعة الثانية. نريد أن نوزِّع علامات (×) بين علامات (•).

إذا كان إجمالي عدد الأشياء يقبل القسمة على عدد علامات ×، فهذا سهل. ما علينا سوى توزيع علامات (×) على علامات • وكأننا نُجري عملية قسمة. على سبيل المثال، إذا كان إجمالي عدد الأشياء يساوي ١٢، منها ثلاثة أشياء تعبِّر عنها العلامة × وتسعة أشياء تعبِّر عنها العلامة •، فإننا نضع علامة × واحدة يليها ثلاث علامات •، ثم علامة × واحدة يليها ثلاث علامات •، وفي النهاية علامة × واحدة يليها ثلاث علامات •:

× • • • × • • • × • • •

لكن ماذا لو كان إجمالي عدد الأشياء — مجموع علامات × وعلامات • — لا يقبل القسمة الصحيحة على عدد علامات ×؟ ماذا يمكن أن نفعل إذا كان لدينا خمس علامات × وسبع علامات •؟

نبدأ بوضع كل علامات × يليها كل علامات • في صف واحد كما يلي:

× × × × × • • • • • • •

ثم نأخذ خمسًا من علامات • ونضعها تحت علامات × كما يلي:

× × × × × • •
• • • • •

نلاحظ في النمط الذي ظهر لنا أنه يتبقى عمودان جهة اليمين. نأخذ العمودين المتبقيين اللذين يشتمل كل واحد منهما على علامة • واحدة، ونضعهما تحت أول عمودين بحيث يشكِّلان صفًّا ثالثًا:

× × × × ×
• • • • •
• •

نلاحظ الآن أنه قد تبقى ثلاثة أعمدة. نأخذ العمودين جهة اليمين ونضعهما تحت العمودين جهة اليسار:

× × ×
• • •
• •
× ×
• •

الآن، لا يتبقى سوى عمود واحد، ومِن ثَم نتوقَّف هنا. نضع الأعمدة في سلسلة تبدأ من اليسار إلى اليمين كما يلي:

× • • × • × • • × • × •

تلك هي النتيجة. لقد وزَّعنا علامات × بين علامات •. لم تتوزَّع العلامات بالتساوي كما في المثال السابق، ولكن تذكَّر أن ذلك مستحيل؛ لأن العدد ١٢ لا يقبل القسمة الصحيحة على العدد ٥. وفي النهاية استطعنا أن نتجنَّب تجميع علامات × بعضها مع بعض وأنشأنا نمطًا لا يبدو عشوائيًّا بالكامل.

قد تتساءل ما إن كان لهذا النمط أيُّ سمة خاصة أم لا؛ قد يفيد هنا إذا وضعنا الصوت «دوم» مكان علامات × والصوت «دا» مكان علامات •. ومِن ثَم يصبح النمط بالصوت «دوم – دا – دا – دوم – دا – دوم – دا – دا – دوم – دا – دوم – دا»، وهذا يعد إيقاعًا بالفعل. يتألَّف الإيقاع من أصواتٍ مشدَّدة يُطلق عليها أيضًا «البادئات» وأصوات غير مشددة أو الأصوات الصامتة. الإيقاع الذي وجدناه ليس إيقاعًا من اختراعنا. فهذا الإيقاع يستخدمه أقزام أكا في جمهورية أفريقيا الوسطى؛ وهو طريقة تصفيق تسمَّى الفيندا في أغنيةٍ من جنوب أفريقيا، كما أنه نمط إيقاعي مستخدَم في مقدونيا في منطقة

البلقان. أزيدكم من الشعر بيتًا. إذا قلبنا النمط، فسيبدأ بعلامة × الثانية (أي بالبادئة)، ومن ثم يصبح بالشكل التالي:

× • × • • × • × • × • •

هذا إيقاع الجرس في كولومبيا ومشهور في كوبا وغرب أفريقيا، وإيقاع طبول في كينيا، ويُستخدم في مقدونيا أيضًا. وإذا قلبناه بحيث يبدأ بالبادئة الثالثة أو الرابعة أو الخامسة، تبرز لنا إيقاعاتٌ أخرى شهيرة حول العالم.

هل نخرج بتلك النتيجة مرة واحدة؟ يمكننا أن نحاول إنشاء إيقاع مكوَّن من ١٢ جزءًا من سبعة أصوات بادئة وخمسة أصوات صامتة؛ وهو نوع من عكس الأصوات البادئة الخمسة والأصوات الصامتة السبعة التي أوضحناها من قبل. إذا اتبعنا الإجراء نفسه بحذافيره، فسنخرج بالنتيجة التالية:

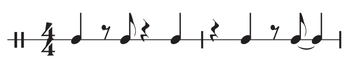
× • × × • × • × × • × •

وهذا إيقاع أيضًا. ويُستخدم في إيقاع إمبيري المشهور في إقليم أشانتي بغانا، وإذا بدأنا الإيقاع بالبادئة الأخيرة، نراه مستخدَمًا بين جماعة اليوروبا في نيجيريا وكذلك في أفريقيا الوسطى وسيراليون.

لئلا تعتقد أننا نسينا بعض الأماكن الجغرافية، إذا بدأنا بخمسة أصوات بادئة و١١ صوتًا صامتًا، نحصل على النمط التالي:

× • • × • • × • • × • • × • • •

هذا إيقاع بوسا نوفا مقلوبٌ. يبدأ إيقاع بوسا نوفا الأصلي بالصوت البادئ الثالث؛ ومِن ثَم فالنمط المطابق المثالي له هو:



× • • × • • × • • • × • • × • •

وإذا بدأنا بثلاثة أصوات بادئة وأربعة أصوات صامتة، فسنحصل على النمط التالي:

× • × • × • •

هذا الإيقاع مشهور في الوزن الإيقاعي سبعة/أربعة، وليس في الموسيقى التقليدية فقط. وهو النمط الإيقاعي لأغنية بينك فلويد «ماني»، من بين نغمات أخرى:

× • × • × • •

يمكن اشتقاق العديد من الإيقاعات الأخرى على هذا النحو عن طريق وضع علامات × وعلامات • في أعمدة ثم تحريكها بالطريقة التي أوضحناها للتو. وقد أوضحنا الإجراء عن طريق قياس الأعمدة المتبقية، ولكن هذه الطريقة تعكس ما يحدث في الواقع. فبدلًا من إنشاء الأعمدة واتباع ما يمليه علم الهندسة، وتحريك الأعمدة، يمكننا فعْل الشيء نفسه بمزيد من المنهجية باستخدام عمليات عددية بسيطة. لتوضيح الأمر، لنرجع إلى مثال الأصوات الصامتة الـ ١٢ والأصوات البادئة السبعة. نبدأ بقسمة ١٢ على ٧، الذي يعطينا خارج القسمة ١ مع تبقي ٥:

$$12 = 1 \times 7 + 5$$

هذا المثال يبيِّن لنا أن نضع سبعة أصوات بادئة في البداية، منشئين بذلك سبعة أعمدة من الأصوات البادئة، يتبعها الأصوات غير المشددة الخمسة المتبقية من عملية القسمة:

×××××××•••••

نعيد القسمة مرة أخرى، ولكن هذه المرة نقسم المقسوم عليه في عملية القسمة السابقة وهو العدد ٧، على الباقي في القسمة السابقة نفسها وهو العدد ٥. ويكون ناتج القسمة ١ مرة أخرى، بينما يكون الباقي ٢:

$$7 = 1 \times 5 + 2$$

وهذا يعني أن علينا أن نأخذ الأعمدة الخمسة جهة اليمين ونضعها تحت الأعمدة الخمسة جهة اليسار، ونترك الباقي وهما ٢:

×××××××
•••••

نكرِّر الخطوة نفسها: نقسم المقسوم عليه في مسألة القسمة السابقة وهو العدد ٥، على الباقي في القسمة السابقة نفسها وهو العدد ٢. فيكون ناتج القسمة ٢ والباقي ١:

$$5 = 2 \times 2 + 1$$

تلك العملية تخبرنا بأن نأخذ «ضعف» العمودين جهة اليمين ونضعهما تحت العمودين جهة اليسار، ونترك الباقي ١:

×××
•••
××
××
••

لاحظ أن كلمة «ضعف» تعني أن هذا يعادل ما كنَّا سنفعله في خطوتين لو اتبعنا طريقة الحل السابقة من دون استخدام عملية القسمة. ومِن ثَم سننتقل من النمط:

×××××××
•••••

إلى النمط التالي أولًا:

×××××
•••••
××

ثم إلى النمط:

×××
•••
××
××
••

وإذا جمعنا الأعمدة في سلسلة، فسنحصل على إيقاع إمبيري:

× • ×× • × • ×× • × •

الخوارزمية الأولى

يمكن كتابة الطريقة التي اتبعناها بمزيد من الدقة بالخطوات التالية. لنفترض أننا نبدأ بعددين وهما العدد a والعدد b ولنفترض أن a يعبِّر عن إجمالي عدد الأصوات. إذا كان عدد الأصوات البادئة أكبر من عدد الأصوات الصامتة، فإن b تعبِّر عن عدد الأصوات البادئة. ولو غير ذلك، فإنها تعبِّر عن الأصوات الصامتة. في البداية، ننشئ صفًّا من الأصوات البادئة، يتبعه صف من الأصوات الصامتة.

(١) نقسم a على b سيكون لدينا ناتج القسمة والباقي. إذا عبَّرنا عن ناتج القسمة بالحرف q وعن الباقي بالحرف r، فستصبح المسألة بالشكل التالي: $a = q \times b + r$. هذه قسمة أعداد صحيحة كما نعرفها. نأخذ q ونضربها في عدد الأعمدة b جهة اليمين ثم ننقلها تحت الأعمدة جهة اليسار، ونترك باقي الأعمدة r جهة اليمين.

(٢) إذا كان الباقي r يساوي صفرًا أو واحدًا، نتوقَّف هنا. أما لو كان غير ذلك، فنعود إلى الخطوة ١، ولكن في هذه المرة ستحل قيمة b مكان قيمة a، وستحل قيمة r مكان قيمة b. أو بعبارة أخرى، سنعود إلى الخطوة ١، ونجعل a تساوي b، ونجعل b تساوي r.

في هاتين الخطوتين، نكرِّر عملية قسمة إلى أن يصبح تكرارها بلا جدوى. يمكنك تتبُّع الخطوات التي اتبعناها في الجدول التالي، حيث نبدأ بـ $a = 12$ و$b = 7$، كما فعلنا من قبل؛ في كل صف يكون شكل المسألة $a = q \times b + r$:

a	q	b	r
12	1	7	5
7	1	5	2
5	2	2	1

إذا أمعنا النظر في الجدول، يمكننا التأكُّد من أن كل صفٍّ يتطابق مع خطوة من خطوات تكوين الصف ونقله، ولكن لدينا تعريف أدق للطريقة التي استخدمناها. في الحقيقة، إن لدينا سلسلة خطوات يمكن إجراؤها بالورقة والقلم، وبذلك تكون هذه

أول خوارزمية معنا! إن لدينا خوارزمية لإنشاء أنماط تتطابق مع العديد من الإيقاعات الموسيقية، والحق أنها كثيرة إلى حدٍّ مذهل. فباستخدام أعداد مختلفة من الأصوات البادئة والأصوات الصامتة، يمكننا الحصول على نحو ٤٠ نمطًا إيقاعيًّا توجد في مختلف الإيقاعات الموسيقية حول العالم. وهنا ينبغي أن نتوقَّف قليلًا: إنها خوارزمية بسيطة (تتكوَّن من خطوتين فقط تتكرَّران)، ومع ذلك يمكن أن تفرز العديد من النتائج المذهلة.

بالرغم من ذلك، بإمكان هذه الخوارزمية أن تفعل ما هو أكثر من ذلك. وما دمنا نتحدَّث عن قسمة عددين، لنفكر في المسألة العامة التالية: إذا كان لدينا عددان، وهما العدد a والعدد b، فما أكبر عدد يقبل القسمة على العددين؟ هذا العدد يسمَّى «العامل المشترك الأكبر» للعددين. لقد تعرَّفنا على العامل المشترك الأكبر في مادة الحساب بالمرحلة الابتدائية في مسائلَ كالمسألة التالية: إذا كان لدينا ١٢ عبوة من الرقائق و٤ عبوات من الجبن، فكيف سنوزِّعها على سلالٍ بحيث يتساوى عدد عبوات الرقائق والجبن في كل سلة؟ بما أن ١٢ تقبل القسمة على ٤، فسيكون لدينا ٤ سلال، تحتوي كل سلَّة منها على ثلاث عبوات من الرقائق وعبوة من الجبن؛ إذن فالعامل المشترك الأكبر للعددين ١٢ و٤ هو ٤. تزداد الأمور إثارةً إذا كان لدينا ١٢ عبوة من الرقائق و٨ عبوات من الجبن. لا يمكنك قسمة عدد على الآخر، ولكن أكبر عدد يقبل العددان ١٢ و٨ القسمة عليه هو ٤، بمعنى أنك ستصنع أربع سلال مرة أخرى تحتوي كلُّ سلة منها على ثلاث عبوات من الرقائق وعبوتين من الجبن.

إذن كيف يمكن إيجاد العامل المشترك الأكبر لأي عددين صحيحين؟ رأينا أنه إذا كان لدينا عددان أحدهما يقبل القسمة على الآخر، فالمقسوم عليه هو العامل المشترك الأكبر. أما إذا لم يقبل أحدهما القسمة على الآخر، فلا نحتاج سوى إيجاد العامل المشترك الأكبر للعدد الباقي من قسمة العددين والعدد الثاني من أجل إيجاد العامل المشترك الأكبر. يسهل فهْم هذا باستخدام الرموز. إذا كان لدينا عددان صحيحان، هما a وb، فالعامل المشترك الأكبر للعدد a والعدد b يساوي العامل المشترك الأكبر لباقي ناتج القسمة $a \div b$ وb. وهذا يعود بنا إلى مسألة الإيقاعات. «الطريقة التي أوجدنا بها الإيقاعات هي نفسها الطريقة التي نستخدمها لإيجاد العامل المشترك الأكبر بين عددين.»

يُطلق على طريقة إيجاد العامل المشترك الأكبر بين عددين «خوارزمية إقليدس» نسبةً إلى عالم الرياضيات اليوناني القديم إقليدس، الذي وصفها للمرة الأولى في كتابه «العناصر» (حوالي سنة ٣٠٠ قبل الميلاد). وتقوم الفكرة الأساسية لها على أن العامل المشترك الأكبر

بين عددين لا يتغيَّر إذا وضعنا العدد الأكبر منهما محلَّ حاصل طرحه من العدد الأصغر. لنضرب مثالًا بالعددين ٥٦ و٢٤. العامل المشترك الأكبر بين العددين هو ٨، وهو أيضًا العامل المشترك الأكبر لحاصل طرح ٥٦ − ٢٤ = ٣٢ والعدد ٢٤، والأمر نفسه ينطبق على العددين ٣٢ و٢٤، وهكذا. إن عملية الطرح المتكرر هي في حقيقتها عملية قسمة؛ ومِن ثَم يمكن وصف خوارزمية إقليدس بالخطوات التالية:

(١) لإيجاد العامل المشترك الأكبر للعددين a وb، نقسم العدد a على العدد b. سيعطينا ذلك ناتج قسمة وباقيًا. إذا عبَّرنا عن ناتج القسمة بالحرف q وعن الباقي بالحرف r، فستصبح المسألة بالشكل التالي: $a = q \times b + r$.

(٢) إذا كان الباقي r يساوي صفرًا، فسنتوقف هنا، وسيصبح العامل المشترك الأكبر للعدد a والعدد b هو b. أما لو كان غير ذلك، فسنعود إلى الخطوة ١، ولكن في هذه المرة ستحل قيمة b مكان قيمة a وستحل قيمة r مكان قيمة b. أو بعبارة أخرى، سنعود إلى الخطوة ١، ونجعل a تساوي b ونجعل b تساوي r.

تلك هي الخطوات التي اتبعناها بالضبط من قبل. الفرق الوحيد هو أننا عند إيجاد الإيقاعات، نتوقَّف عندما يصبح الباقي ٠ أو ١ في الخطوة الثانية، بينما تتوقَّف خوارزمية إقليدس عندما يكون الباقي صفرًا. إن الأمر سيان في الواقع؛ فإذا كان الباقي يساوي ١، فعند تكرار الخطوة ١ مرة أخرى سيكون الباقي صفرًا؛ لأن أيَّ عدد صحيح يقبل القسمة على ١. لنجرِّب العددين ٩ و٥: ٩ = ١ × ٥ + ٤، ومِن ثَم ننتقل إلى ٥ = ١ × ٤ + ١، ثم إلى ٤ = ١ × ٤ + ٠، وبذلك يكون العامل المشترك الأكبر بين العددين ٩ و٥ هو ١.

قد يكون من المفيد رؤية الخوارزمية في مثال عملي حيث $a = 136$، و$b = 56$ في الجدول التالي، المماثل للجدول الذي رأيناه من قبل في مثال الإيقاعات. نجد أن العامل المشترك الأكبر بين العددين ١٣٦ و٥٦ هو العدد ٨:

a	q	b	r
136	2	56	24
56	2	24	8
24	3	8	0

كما أشرنا في مثال العددين ٩ و٥، تعمل خوارزمية إقليدس على نحو صحيح في جميع الحالات، حتى عندما لا يكون بين العددين أيُّ عامل مشترك غير العدد ١. هذا ما حدث مع المثال $a = 9$ و$b = 5$. يمكنك أن ترى بنفسك ما يحدث إذا جرَّبت تطبيق خطوات الخوارزمية مع المثال $a = 55$ و$b = 34$؛ سيستغرق هذا المثال بضع خطوات، ولكن ستنتهي الخوارزمية إلى أن العامل المشترك الوحيد هو العدد ١.

تنفَّذ خطوات خوارزمية إقليدس بترتيب محدَّد ودقيق. ويوضح وصف الخوارزمية الطريقة التي تندمج بها خطواتها:

(١) توضع الخطوات في «تسلسل».
(٢) قد تبيِّن الخطوات «اختيارًا» يحدِّد الخطوات التي ينبغي اتباعها. في الخطوة ٢، نجد اختبارًا لمعرفة إذا ما كان الباقي يساوي صفرًا أم لا. ومِن ثَم يصبح لدينا بديلان بناءً على الناتج: إما التوقُّف أو العودة إلى الخطوة ١.
(٣) يمكن وضع الخطوات في «حلقة» أو «تكرار» حيث يتكرَّر تنفيذها. في الخطوة ٢، إذا لم يكن الباقي يساوي صفرًا، نعود إلى الخطوة ١.

نطلق على تلك الطرق الثلاث لدمج الخطوات اسم «بنيات التحكُّم»؛ لأنها تملي الإجراء الذي سيُتخذ عند تطبيق الخوارزمية. وجميع الخوارزميات تُبنَى بتلك الطريقة. إنها تتكوَّن من خطوات لإجراء العمليات الحسابية ومعالجة البيانات؛ إذ تُجمع هذه الخطوات معًا وتُصمَّم باستخدام بنيات التحكم الثلاث المذكورة. وكلما زاد تعقيد الخوارزمية، زادت خطواتها وربما زاد تعقيد تصميمها كذلك. لكن بنيات التحكم الثلاث كافية لوصف الطريقة التي ينبغي بها دمج خطوات الخوارزمية معًا.

تعمل خطوات الخوارزمية — من بين أشياء أخرى — بناءً على المدخلات التي نوفِّرها. والمدخلات هي البيانات التي تعالجها الخوارزمية. وإذا اعتمدنا طريقة عرض تتمحور حول البيانات، فسنستخدم خوارزمية لتحويل بعض البيانات — التي تصف مسألةً ما — إلى شكلٍ يتطابق مع حل المسألة.

لقد وجدنا خوارزمية وراء الإيقاعات الموسيقية عبارة عن تطبيق لعملية قسمة، ولكن في الواقع لا داعي إلى الذهاب إلى بعيد؛ فالقسمة في حد ذاتها عبارة عن خوارزمية. حتى إن لم تسمع عن خوارزمية إقليدس، فأنت تعرف كيفية قسمة عددين كبيرين؛ كلنا قضينا بعض الوقت خلال سنوات التعلُّم الأولى في إجراء مسائل ضرب وقسمة مطوَّلة. وقضى

معلمونا ساعاتٍ يحفرون طريقةَ حل تلك المسائل في عقولنا: إنها مجموعة خطوات نضع فيها الأعداد في الخانات في المواضع الصحيحة ونجري العمليات الحسابية بها، وتلك هي الخوارزميات. ولكن الخوارزميات ليست مرتبطة بالأعداد فحسب كما رأينا منذ لحظات. فقد اكتشفنا أنها مرتبطة بطريقة إعداد إيقاع موسيقي. ومع ذلك فلا يوجد شيء محيِّر بشأنها. فالإيقاع هو طريقة لتوزيع النبرات في فترة زمنية معينة، وينطبق المبدأ نفسه عند تعبئة الرقائق والجبن.

كان لتطبيق خوارزمية إقليدس على الإيقاعات مصدر غير محتمل، ألا وهو منشأة بها «مصدر نيوترونات» في مختبر أوك ريدج الوطني بولاية تينيسي. ينتج مصدر التشظي النيوتروني (SNS) في هذا المختبر أشعة نيوترونات نابضة قوية تُستخدم في تجارب فيزياء الجسيمات. (الفعل «تشظَّى» يعني انقسام المادة إلى أجزاء أصغر؛ وفي الفيزياء النووية، توجد نواة ثقيلة تصدر عددًا كبيرًا من البروتونات بعد تفجيرها باستخدام جسيم ذي طاقة عالية.) خلال تشغيل مصدر التشظي النيوتروني، يجب تشغيل بعض العناصر — مثل مصادر توريد الطاقة العالية الجهد — بحيث تُوزَّع النبضات في فسحاتٍ زمنية بالتساوي قدْر الإمكان. والخوارزمية التي ابتُكرت من أجل عملية التوزيع لا تختلف في جوهرها عن خوارزمية صنْع الإيقاعات وخوارزمية إقليدس؛ آخذة إيانا من الأعداد إلى الجسيمات دون الذرية إلى الإيقاع الموسيقي.[3]

الخوارزميات وأجهزة الكمبيوتر والرياضيات

قلنا إن الخوارزميات ليست مرتبطة بأجهزة الكمبيوتر، على الرغم من أن الغالبية يربطون بينهما في وقتنا الحالي. صحيح أن الخوارزميات تُظهر إمكانياتها عندما تقترن بأجهزة الكمبيوتر، ولكن الكمبيوتر ما هو في الواقع إلا آلة تملك تلك السمة الخاصة التي تمكننا من إعطائه أوامرَ لإنجاز مهامَّ بعينها. ونحن نعطي تلك الأوامر عن طريق «البرمجة»، وعادةً ما نقوم ببرمجته من أجل تنفيذ الخوارزميات.

هذا التوضيح يقودنا إلى البرمجة نفسها. البرمجة هي نظامٌ لتحويل أهدافنا إلى رموزٍ يستطيع جهاز الكمبيوتر فهْمها. نطلق على تلك الرموز «لغة البرمجة»؛ لأنها في بعض الأحيان تشبه ما نكتبه بلغة بشرية، ولكن لغات البرمجة مسألة بسيطة للغاية مقارنةً بثراء اللغات البشرية وتعقيدها. في الوقت الحاضر، لا يفهم جهاز الكمبيوتر أيَّ شيء بالطبع. قد تتغيَّر الأمور في المستقبل إذا استطعنا أن ننتج آلاتٍ ذكية بحق، ولكن في الوقت

الحاضر عندما نقول إن الكمبيوتر يفهم الرموز، فهذا يعني في الحقيقة أن الرموز تتحوَّل إلى سلسلةٍ من التعليمات لمعالجة التيار في الدوائر الإلكترونية (يمكن أيضًا استخدام التيار الخفيف بدلًا من التيار الكهربي، وإن كانت الفكرة واحدة).

البرمجة هي نظامٌ لتحويل أهدافنا إلى رموزٍ يستطيع جهاز الكمبيوتر فهْمها. نطلق على تلك الرموز «لغة البرمجة».

إذا كانت الخوارزمية مجموعة خطوات يمكننا تنفيذها بأنفسنا، فالبرمجة هي النشاط الذي ندوِّن به الخطوات بالرموز التي يفهمها الكمبيوتر. وعندئذٍ يكون الكمبيوتر هو من سيقوم بتنفيذ تلك الخطوات. فأجهزة الكمبيوتر أسرع بكثير من البشر؛ ولذا يمكن أن تنفِّذ الخطوات في وقتٍ أقل. والعامل الأساسي في الحوسبة هو «السرعة». أما من الناحية النوعية، فلا يمكن لجهاز الكمبيوتر أن يفعل أكثر مما يفعله البشر، ولكنه يقوم به على نحوٍ أسرع، بل أسرع كثيرًا. فالخوارزمية تكتسب قوة على الكمبيوتر نظرًا لإمكانية تنفيذها عليه في جزء صغير من الوقت الذي يستغرقه الإنسان كي ينفذ الخطوات نفسها، «لكن في النهاية تظل الخطوات واحدة».

إذا كانت الخوارزمية مجموعة خطوات يمكننا تنفيذها بأنفسنا، فالبرمجة هي النشاط الذي ندوِّن به الخطوات بالرموز التي يفهمها الكمبيوتر.

توفِّر لنا لغة البرمجة طريقةً لوصف خطوات تنفيذ الخوارزميات لجهاز الكمبيوتر. كذلك توفِّر وسيلةً لبناء تلك الخوارزميات باستخدام ثلاث بنيات تحكُّم أساسية وهي: التسلسل والاختيار والتكرار. فنحن نكتب الخطوات ونصف طريقة تصميمها باستخدام المفردات والعبارات التي توفِّرها لغة البرمجة التي نستخدمها.

توجد ميزة إضافية لاستخدام أجهزة الكمبيوتر غير السرعة؛ إن كنت تتذكَّر كيف تعلمتَ طرق حل مسائل الضرب والقسمة المطولة، فستجد أن الأمر ربما استغرق تدريبًا لفترة طويلة، وربما لم يكن مسليًا. وكما أشرنا سلفًا، تُحفر هذه الدروس داخل عقولنا في عمرٍ مبكِّر، وحفر الأشياء في العقل ليس بالأمر السار. أما أجهزة الكمبيوتر فلا تعاني السَّأم؛ ومِن ثَم يكون من الأسباب الأخرى لترك مهمة تنفيذ الخوارزميات لها أن تزيح عنا الملل وتترك لنا وقتًا للقيام بمهامَّ أكثر إثارة ومتعة.

على الرغم من أن الخوارزميات عادة ما تنفَّذ على جهاز كمبيوتر، بعد كتابتها بإحدى لغات البرمجة، فإنها في الأساس مكتوبة للبشر؛ الذين ينبغي أن يفهموا آلية عملها ومتى يمكن استخدامها. وهذا يقودنا إلى شيء أساسي يغفل عنه حتى علماء الكمبيوتر المتمرسون والمبرمجون المحنَّكون. هذا الشيء هو أن الطريقة الوحيدة لفهم الخوارزمية بحق هو تنفيذها يدويًّا. لا بد أن نتمكَّن من تنفيذ الخوارزمية بالطريقة نفسها التي ينفِّذ بها جهاز الكمبيوتر البرنامج الذي يطبِّقها. وفي الوقت الحاضر، نحظى بمجموعة مذهلة من ملفات الوسائط بين أيدينا يمكن أن تساعدنا في التعلم، مثل تطبيقات المحاكاة والرسوم المتحركة ومقاطع الفيديو الرائعة التي يمكن الوصول إليها بضغطة زر واحدة. كل هذا رائع، ولكن عندما يستعصي عليك أمرٌ، احتفظ بالورقة والقلم بجانبك. ينطبق الأمر نفسه على هذا الكتاب. هل فهمت حقًّا طريقةَ إنشاء الإيقاعات؟ هل جرَّبت إنشاء إيقاع؟ هل يمكنك إيجاد العامل المشترك الأكبر بين العددين ٢٥٢ و٢٤؟

كل البرامج تنفِّذ مجموعة من الخطوات لتنفيذِ مهمةٍ ما، ومِن ثَم يمكننا القول إن كل البرامج عبارة عن خوارزميات. ولكننا نتمتع بقدرٍ أكبر قليلًا من الدقة ونريد لخطواتنا أن تلبي خصائص معينة:[4]

(١) يجب أن تنتهي الخطوات بعد عدد محدَّد من الخطوات. فلا يمكن لخوارزمية أن تستمر إلى ما لا نهاية. (يمكن لبرنامجٍ أن يعمل بلا انقطاعٍ ما دام جهاز الكمبيوتر الذي يعمل عليه قيد التشغيل. وفي تلك الحالة لا يكون البرنامج تنفيذًا لخوارزمية، بل فقط مجرد عملية حاسوبية.)

(٢) يجب أن تكون الخطوات دقيقة بحيث يمكن تنفيذها دون ارتباك أو حيرة.

(٣) قد تعمل الخوارزمية على بعض المدخلات؛ كخوارزمية إقليدس التي تعمل على عددين صحيحين.

(٤) للخوارزمية بعض النتائج؛ وهذا مجمل الهدف منها: أن تنتج شيئًا يهدف إلى الوصول إلى نتيجة. والنتيجة في خوارزمية إقليدس هي العامل المشترك الأكبر.

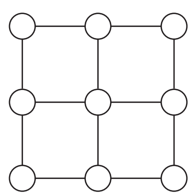
(٥) يجب أن تكون الخوارزمية فعَّالة. يجب أن يتمكَّن الإنسان من تنفيذ كل خطوة في قدرٍ معقول من الوقت باستخدام الورقة والقلم.

تضمن هذه الخصائص الوصول إلى نتيجةٍ من الخوارزمية. فوجود الخوارزميات يُعزى إلى قيامها بشيء مفيد. توجد خوارزميات عبثية وقد يخترع علماء الكمبيوتر

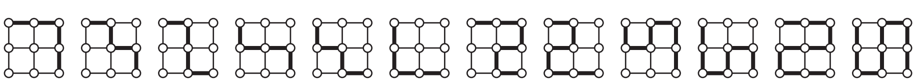
خوارزميات لا فائدة منها، إما على سبيل المزاح أو بالخطأ، ولكننا في الحقيقة مهتمون بالخوارزميات التي تحمل بعض الفائدة لنا. عند العمل بالخوارزميات، لا يكفي أن تبيِّن قدرتها على إنجاز مهمةٍ ما. نريد أن تكون الخوارزمية ذات فائدة عملية، ولذا يجب أن تؤدي إلى نتيجة جيدة.

هنا يكمن فارق جوهري بين الخوارزميات والرياضيات. معظم علماء الكمبيوتر الأوائل كانوا علماء في الرياضيات، وعلم الكمبيوتر يستخدم كمًّا هائلًا من الرياضيات، ولكنه ليس منهجًا رياضيًّا. فعالِم الرياضيات يريد إثبات «نتيجةٍ ما»؛ أما عالِم الكمبيوتر فيريد لتلك النتيجة «أن تؤتي ثمارها».

الخاصية الأولى للخوارزميات هي ضرورة التقيُّد بعددٍ محدَّد من الخطوات. تلك الخاصية ليست دقيقة للغاية. فنحن لا نريد مجرد التقيُّد بعدد محدَّد من الخطوات. بل نريد عددًا قليلًا من الخطوات يكفي لتنفيذها عمليًّا، بحيث تنتهي الخوارزمية في غضون فترة زمنية معقولة. وهذا يعني أن مجرد التوصُّل إلى خوارزمية غير كافٍ؛ بل يجب أن تكون فعَّالة من المنظور العملي. لنضرب مثالًا كي نوضِّح الفرق بين معرفة الشيء ومعرفة طريقة عمل الشيء بكفاءة. تخيَّل أن لدينا شبكة كالتالية:



نريد إيجاد أقصر مسار من الركن العلوي جهة اليسار إلى الركن السفلي جهة اليمين من دون المرور على مكان واحد مرتين. طول كل مسار يساوي عدد الخطوط التي تربط بين النقاط على الشبكة. توجد هنا طريقة واحدة للقيام بذلك وهي: إيجاد جميع تلك المسارات، وقياس طول كل مسار، واختيار أقصرها، أو اختيار أيٍّ من المسارات الأقصر طولًا حال وجود أكثر من واحد. إجمالي عدد المسارات يساوي ١٢، كما نرى فيما يلي:



توجد ستة مسارات بطول ٤، ومِن ثَم يمكننا اختيار أي مسار منها.

لكننا لسنا مقيدين بشبكاتٍ تتكوَّن من ٣ × ٣ نقاط. يمكن أن يكون لدينا شبكات مكونة من ٤ × ٤ نقاط أو ٥ × ٥ نقاط أو حتى أكبر. ومِن ثَم نكتشف أن تلك الطريقة لا تصلح في جميع الحالات. يوجد ١٨٤ مسارًا من الركن العلوي جهة اليسار إلى الركن السفلي جهة اليمين في شبكة تتكوَّن من ٤ × ٤ نقاط؛ وإذا انتقلنا إلى شبكة تتكوَّن من ٥ × ٥ نقاط، فسيزيد عدد المسارات إلى ٨٥١٢ مسارًا. يستمر عدد المسارات في الزيادة بسرعة — بل يزيد بخطًى متسارعة باستمرار — وحتى عدُّ تلك المسارات أمرٌ صعب. فعندما نصل إلى شبكة مكونة من ٢٦ × ٢٦ نقطة، يصبح لدينا ٨ ٤٠٢ ٩٧٤ ٨٥٧ ٨٨١ ١٣٣ ٤٧١ ٠٠٧ ٠٨٣ ٧٤٥ ٤٣٦ ٨٠٩ ١٢٧ ٢٩٦ ٠٥٤ ٢٩٣ ٧٧٥ ٣٨٣ ٥٤٩ ٨٢٤ ٧٤٢ ٦٢٣ ٩٣٧ ٠٢٨ ٤٩٧ ٨٩٨ ٢١٥ ٢٥٦ ٩٢٩ ١٧٨ ٥٧٧ ٠٨٣ ٩٧٠ ٩٦٠ ١٢١ ٦٢٥ ٦٠٢ ٥٠٦ ٠٢٧ ٣١٦ ٥٤٩ ٧١٨ ٤٠٢ ١٠٦ ٤٩٤ ٠٤٩ ٩٧٨ ٣٧٥ ٦٠٤ ٢٤٧ ٤٠٨ من المسارات. يحتوي هذا الرقم على ١٥١ عددًا عشريًّا، ووُجد هذا الناتج باستخدام برنامج كمبيوتر يستخدم خوارزمية؛ نعم، نحن نستخدم خوارزمية لفهْم سلوك خوارزمية أخرى.[5]

إن طريقة عدِّ جميع المسارات واختيار المسار الأقصر منها صحيحةٌ بلا شك، وستعطينا المسار الأقصر دائمًا — أو أيًّا من المسارات الأقصر إذا كان هناك عدد من المسارات المتساوية القصر — ولكن تلك الطريقة ليست عملية بالتأكيد. كما أنها ليست مفيدة على الإطلاق؛ إذ توجد خوارزميات ستوجِد أقصر مسار دون الاضطرار إلى عدِّ كل المسارات المحتملة، ومِن ثَم توفِّر قدرًا كبيرًا من الوقت، وتتيح لنا التعامل مع الشبكات مهما كان حجمها. ففي الشبكة التي تتكوَّن من ٢٦ × ٢٦ نقطة، يصل عدد الخطوات اللازمة للحصول على الإجابة إلى مئات تقريبًا؛ سنرى ذلك في الفصل التالي.

إن السؤال عن ماهية الخوارزمية العملية وبأي معنًى تكون الخوارزمية عملية أكثرَ من غيرها هو من صميم أي تطبيق للخوارزميات. وسنرى على مدى ما تبقى من الكتاب أنه كثيرًا ما توجد خوارزميات مختلفة لحل مسألة واحدة، ونحن مَن نختار الخوارزمية الأنسب للتطبيق في كل موقف بعينه. فمثل جميع الأدوات، بعض الخوارزميات تكون أنسب لحالاتٍ معينة من غيرها. ولكن على خلاف العديد من الأدوات الأخرى، نمتلك طريقة محدَّدة لتقييم مزايا الخوارزميات.

قياس الخوارزميات

عندما نبحث في خوارزمية لحل مسألة ما، نرغب في معرفة الكيفية التي ستقوم بإجراء المسألة من خلالها. ودائمًا ما تكون السرعة عاملًا مهمًّا في هذا الشأن. فنحن نستخدم الخوارزميات على الكمبيوتر لإنجاز المهام أسرع من الإنسان.

مع تحسُّن العناصر المكوِّنة لجهاز الكمبيوتر، عادةً ما لا نكتفي بمعرفة طريقة عمل البرنامج الذي ينفِّذ خوارزمية ما على كمبيوتر معيَّن. فقد يكون جهاز الكمبيوتر الخاص بنا أسرع أو أبطأ من الكمبيوتر الذي قيست عليه الخوارزمية، وبعد مرور بضع سنوات، لن يكون لقياسات الخوارزميات على الأجهزة القديمة أي أهمية سوى الأهمية التاريخية. نحن بحاجة إلى معرفة مدى جودة أداء الخوارزمية بعيدًا عن مكوِّنات الكمبيوتر.

لكن ينبغي أن ينعكس حجم المسألة التي نحاول حلها في طريقة قياس أداء الخوارزمية. فنحن في الحقيقة لا يهمنا الوقت المستغرق في ترتيب ١٠ عناصر؛ فبإمكاننا أن نفعل ذلك بأيدينا على أي حال. بل يهمنا الوقت المستغرق في ترتيب مليون عنصر أو أكثر. نحن نريد مقياسًا لتوقعنا لأداء الخوارزمية في المسائل غير التافهة.

وفي سبيل ذلك، نحتاج إلى طريقةٍ لتحديد حجم المسائل التي تُغذَّى بها الخوارزمية. ويتفاوت بُعد الأهمية بين مختلف المسائل. فإذا أردنا فرز عدد من العناصر على الكمبيوتر، فالبعد المهم هو عدد العناصر التي نريد فرزها (وليس حجم العناصر أو تكوينها مثلًا). وإذا أردنا ضرب عددين، فالبعد المهم هو عدد الأرقام في العددين (هذا البُعد يهم الإنسان أيضًا؛ لأن السبب في «طول» عملية الضرب المطولة راجع إلى اعتمادها على عدد الأرقام في كل عدد من العددين). عندما ندرس مسألة ونُرشِّح خوارزمية لحلها، فإننا نفعل ذلك دومًا واضعين في الاعتبار حجم المسألة.

على الرغم من تنوُّع طرق تقييم حجم مسائل بعينها، فإننا في النهاية نحدِّد حجم كل مسألة بعدد صحيح، نطلق عليه n. بالعودة إلى الأمثلة السابقة، فإن n يمثِّل إما عدد العناصر المطلوب فرزُها أو عدد الأرقام في العددين المراد ضربهما. إذن، ما نريده هو أن نكون قادرين على التحدُّث عن أداء الخوارزميات التي تحل مسائل بحجم n.

يرتبط الوقت الذي تحتاج إليه الخوارزمية بما تتسم به من «تعقيد حسابي». والتعقيد الحسابي للخوارزمية هو مقدار الموارد اللازمة كي تعمل. يوجد نوعان أساسيان من الموارد هنا هما: الوقت، أي المدة التي تستغرقها الخوارزمية، والمساحة، أي السعة اللازمة لها في ذاكرة التخزين الخاصة بالكمبيوتر.

سنركِّز في الوقت الحالي على الوقت. نظرًا لتنوُّع أجهزة الكمبيوتر تبعًا لتنوُّع مواصفات الأداء فيها؛ فالحديث عن الوقت الذي تستغرقه الخوارزمية لكي تعمل على جهاز كمبيوترٍ معيَّن قد يعطينا مؤشرًا لما يفترض أن نتوقَّعه عند تشغيلها على أجهزة كمبيوتر أخرى، ولكننا نريد شيئًا أعمَّ. تعتمد سرعة الكمبيوتر على الوقت الذي يستغرقه في إجراء عمليات أساسية. ولتجنُّب مثل تلك التفاصيل، سنختار الحديث عن «عدد العمليات» اللازمة لتشغيل خوارزمية ما، وليس عن الوقت الفعلي الذي تستغرقه الخوارزمية على كمبيوتر معيَّن لإجراء تلك العمليات.

بعد التوضيح، يُرجى العلم أننا سنسيء استعمال المصطلحات قليلًا ونعامل لفظة «العمليات» ولفظة «الوقت» باعتبارهما مترادفتين. على الرغم من ضرورة التزام الدقة عند القول إن الخوارزمية تتطلب «العدد س من العمليات»، سنقول أيضًا إن الخوارزمية تستغرق «المدة س من الوقت» للإشارة إلى أنها تعمل في المدة اللازمة لتنفيذ العدد س من العمليات على أي كمبيوتر تعمل عليه الخوارزمية فعليًّا. وعلى الرغم من تفاوت الوقت الفعلي المطلوب باختلاف مكوِّنات الكمبيوتر، فإن هذا لا يهم عندما نريد المقارنة بين خوارزميتين تعملان في «الوقت س» و«الوقت ص» على جهاز الكمبيوتر «نفسه» بغض النظر عن مواصفاته.

نعود الآن إلى حجم المسألة المدخلة إلى الخوارزمية. بما أننا مهتمون بالمسائل غير التافهة، فلن نأبه بما يحدث مع المسائل الصغيرة الحجم. بل سنهتم بما يحدث في المسألة بمجرد الوصول إلى حجم معيَّن. لن نحدد حجم تلك المسائل بالضبط، ولكننا سنفترض دومًا أنها كبيرة.

يوجد تعريف للتعقيد ثبتت فائدته من الناحية العملية. وهو يحتوي أيضًا على رمز واسم. نكتب $O\left(\cdot\right)$ ونسميه ترميز *Big O*. داخل ترميز Big O في مكان النقطة، نكتب مقدارًا جبريًّا. يعني الرمز أن الخوارزمية ستستغرق وقتًا يساوي مضاعف المقدار الجبري في الأغلب. لنرَ ما يعنيه ذلك:

- إذا كنت تريد البحث عن شيء في سلسلة عناصر — يوجد عدد n من العناصر — وسلسلة العناصر ليست مرتَّبة بأي صورة، فإن التعقيد يساوي $O\left(n\right)$. وهذا يعني، بالنسبة إلى العدد n من العناصر، أن الوقت اللازم لإيجاد عنصر معيَّن في تلك السلسلة لن يزيد على مضاعف ضرب عدد العناصر.

- إذا كنت تريد ضرب عددين مكوَّنين من n من الأرقام باستخدام عملية ضرب مطولة، فإن التعقيد يساوي $O\left(n^2\right)$. وهذا يعني أن الوقت اللازم لإنجاز عملية الضرب لن يزيد على مضاعف مربع حجم العددين.

إذا كان لدينا خوارزمية درجة تعقيدها $O\left(n\right)$، فإننا نتوقَّع لمدخل بحجم ١٠٠٠٠ أن يحتاج إلى مضاعف عشرة آلاف خطوة. وإذا كانت درجة التعقيد في الخوارزمية تساوي $O\left(n^2\right)$، فإننا نتوقَّع، بالنسبة إلى حجم مدخل مساوٍ للسابق، أن تحتاج إلى مائة مليون خطوة. ولا يعتبر هذا الحجم كبيرًا بالنسبة إلى العديد من المسائل. فعادةً ما ترتب أجهزة الكمبيوتر ١٠٠٠٠ عنصر. ولكنك ترى أن نطاق عدد الخطوات الذي يمثله تعقيد الخوارزمية يمكن أن يزيد.

فيما يلي بعض الأمثلة التي قد تساعدك في تقدير حجم بعض الأرقام التي سنتعرض لها. لنأخذ العدد ١٠٠ مليار، أو $١٠^{١١}$، الذي يتكوَّن من الرقم ١ وخلفه ١١ صفرًا. إذا أخذت ١٠٠ مليار شطيرة هامبرجر ووضعتها متلاصقة بعضها بجانب بعض، يمكنك أن تلف الكرة الأرضية ٢١٦ مرة وتصل إلى القمر وتعود مرة أخرى.

عادة ما يُطلق على المليار من الشيء «جيجا»، على الأقل في أجهزة الكمبيوتر. بعد المليار — أو الجيجا — يأتي التريليون — أو «التيرا» — الذي يساوي ١٠٠٠ مليار، أو $١٠^{١٢}$. إذا بدأت العدَّ بعددٍ واحد في الثانية، فستحتاج إلى ٣١٠٠٠ سنة كي تصل إلى تريليون واحد. بالضرب في ١٠٠٠ أخرى، نصل إلى الكوادريليون، أو $١٠^{١٥}$، أو «البيتا»؛ يتراوح إجمالي النمل الذي يعيش على الأرض بين ١ و١٠ كوادريليون نملة، على حد قول عالِم الأحياء إي أو ويلسون. بعبارة أخرى، يتراوح عدد النمل على كوكب الأرض بين ١ و١٠ بيتا نملة.

بعد الكوادريليون، يأتي الكوينتليون أو «إكسا»؛ يساوي الكوينتليون $١٠^{١٨}$ وهو العدد التقريبي لحبيبات الرمال في ١٠ شواطئ كبرى. على سبيل المثال، تحتوي عشرة شواطئ بحجم شاطئ كوباكابانا على واحد إكسا من حبات الرمال. بالضرب مرة أخرى، نصل إلى $١٠^{٢١}$، أو واحد سكستليون، أو «زيتا». يبلغ عدد النجوم في الكون المرئي لنا واحد زيتا نجم. ننتهي من البادئات بعد «يوتا»، التي ترمز إلى $١٠^{٢٤}$، أي واحد سيبتليون. لكن الأعداد لا تتوقَّف عن الزيادة. فيُطلق على العدد $١٠^{١٠٠}$ «جوجول» — نعم، ربما تعرف شركة أطلقت على نفسها اسم هذا العدد بخطأ إملائي مقصود. بعد ذلك يأتي العدد ١٠ مرفوعًا إلى مقام الأُس للعدد جوجول — $١٠^{١٠^{١٠٠}}$ — ويساوي واحد جوجولبليكس.[6]

ستساعدنا تلك الأمثلة في تقدير المزايا النسبية لخوارزمياتٍ معينة سنتناولها فيما تبقَّى من هذا الكتاب. وعلى الرغم من إمكانية وجود خوارزميات — من الناحية النظرية — تنطوي على أي نوع من التعقيد، فإن الخوارزميات التي نتعامل معها عادةً ما تندرج تحت عدد قليل من المجموعات المختلفة.

فئات التعقيد

تتألَّف أسرع فئة بين جميع فئات الخوارزميات من الخوارزميات التي تعمل في وقتٍ لا يتعدى الوقت الثابت، مهما كان حجم المدخلات. ونشير إلى هذا التعقيد بالرمز $O\left(1\right)$؛ على سبيل المثال، خوارزمية تتحقَّق إن كان الرقم الأخير في عددٍ ما فرديًّا أو زوجيًّا لن تتأثر بحجم العدد وستعمل خلال الوقت الثابت. الرقم ١ في الرمز $O\left(1\right)$ نابع من حقيقة أن الرمز $O\left(1\right)$ يعني أن الخوارزمية لا تحتاج في تشغيلها إلى أكثر من مضاعف خطوة واحدة؛ أي عدد ثابت من الخطوات.

قبل أن نتناول فئة التعقيد التالية، نحتاج إلى الاطلاع سريعًا على طريقةٍ خاصة يمكن من خلالها أن تزيد العناصر أو تتقلص. إذا جمعت شيئًا مع نفسه عدة مرات، فأنت بذلك تضاعفه. وإذا ضربت شيئًا في نفسه عدة مرات، فأنت بذلك ترفعه إلى قوة أسية. وقد رأينا لتونا مدى الضخامة التي يمكن أن تصل إليها الأرقام ذات الأس، مثل $١٠^{١٢}$ (أو أكبر). ما قد لا يتضح على الفور هو مدى السرعة التي تؤدي بها عملية الرفع إلى الأس إلى تصاعد مذهل، وتلك ظاهرة تسمَّى «النمو الأسي».

سوف تتضح تلك النقطة من خلال قصة اختراع الشطرنج التي يرجَّح أن تكون ملفَّقة. طلب حاكم البلد الذي اختُرِع فيه الشطرنج من مخترعه أن يطلب أيَّ هدية يريدها (للأسف في هذه القصص يكون البطل رجلًا). فأجاب أنه يريد حبَّة واحدة من الأرز في المربع الأول من لوحة الشطرنج، وحبتين في المربع الثاني، وأربعًا في المربع الثالث، وهكذا. ظن الملك أنه قد أفلت من الأمنية بسهولة وأعطاه ما أراد. ولكن للأسف، سرعان ما ساءت الأمور. فالسلسلة تزيد بمعدل أس ٢: $٢^{٠} = ١$ في المربع الأول، $٢^{١} = ٢$ في المربع الثاني، $٢^{٢} = ٤$ في المربع الثالث، ومن ثم سيبلغ عدد الحبات في المربع الأخير $٢^{٦٣}$، وهو كمٌّ لا يمكن بلوغه بأي حال (إذ يساوي ٩,٢٢٣,٣٧٢,٠٣٦,٨٥٤,٧٧٥,٨٠٨ أو ما يقرب من ٩ كوينتليون).

يمكن أن يساعدنا النمو الأسي أيضًا في فهْم السبب وراء الصعوبة البالغة في طي قطعة من الورق عدة مرات. ففي كل مرة تطوي فيها الورقة، يتضاعف عدد طبقات الورقة المطوية. وبعد عشر طيات، ستحصل على ٢^{١٠} = ١٠٢٤ طبقة. وإذا كان سُمك الورقة ٠٫١ ملِّيمتر، فستحصل على رزمة مطوية يزيد سُمكها على ١٠ سنتيمترات. بصرف النظر عن القوة التي ستحتاج إليها لطي الورقة إلى طبقتين؛ فقد يستحيل طيها من المنظور الفيزيائي؛ لأن طول الشيء لا بد أن يكون أكبر من سُمكه حتى يمكن طيه.[7]

النمو الأسي هو السبب في زيادة قدرات أجهزة الكمبيوتر أكثر وأكثر على مدى السنين. فطبقًا لقانون مور، يتضاعف عدد الترانزستورات في الدائرة المدمجة كل عامين تقريبًا. وقد سُمي القانون بهذا الاسم نسبة إلى جوردون مور الذي أسَّس شركتي فيرتشايلد لأشباه الموصلات وإنتل. وقد قدَّم تلك الملاحظة في عام ١٩٦٥؛ وثبتت صحة القانون، ومِن ثَم انتقلنا من نحو ٢٠٠٠ ترانزستور في المعالج عام ١٩٧١ (معالج Intel 4004) إلى أكثر من ١٩ مليار ترانزستور في عام ٢٠١٧ (معالج 32-core AMD Epyc).[8]

بعدما اطلعنا على النمو، لنتناول العكس الآن. إذا كان لديك حاصل ضرب شيءٍ ما، فستستخدم القسمة لعكس العملية والحصول على القيمة الأصلية. إذا كان لديك القوة الأسية لشيءٍ ما، a^n، فكيف لنا أن نعكس العملية؟ إن معكوس الرفع إلى الأس هو «اللوغاريتم».

تُعتبر اللوغاريتمات في بعض الأحيان الحد الفاصل بين الرياضيات للجميع والرياضيات للمبتدئين؛ حتى الاسم تحوطه هالةٌ من عدم الفهم. إذا بدت اللوغاريتمات غامضة بعض الشيء، فعليك أن تتذكَّر أن لوغاريتم العدد هو معكوس رفع العدد إلى قوة أسية. فكما في رفع العدد إلى قوة أسية تكرار لعملية الضرب، فإن في تناول لوغاريتم ما تكرارًا لعملية القسمة.

اللوغاريتم هو الإجابة على سؤال: «إلى أي قوة أسية ينبغي أن أرفع العدد كي أحصل على القيمة التي أريدها؟» والعدد الذي نرفعه يسمَّى «أساس» اللوغاريتم. فإذا كان السؤال هو: «إلى أي قوة أسية ينبغي أن أرفع العدد ١٠ للحصول على العدد ١٠٠٠؟» فستكون الإجابة هي ٣؛ لأن ١٠^٣ = ١٠٠٠. بالطبع قد نريد رفع عدد مختلف، بمعنى أن نستخدم أساسًا مختلفًا. رمز اللوغاريتمات هو $log_a x$، وهذا يتطابق مع السؤال: «إلى أي قوة أسية ينبغي أن أرفع العدد a كي أحصل على النتيجة x؟» عندما تكون $a = 10$، فإننا نسقط الرمز المنخفض فقط؛ لأن الأساس ١٠ مشترك في اللوغاريتمات، وبدلًا من كتابة $log_{10} x$، نكتب $log x$ ببساطة.

يوجد أيضًا أساسان مشتركان آخران. عندما يكون الأساس هو الثابت الرياضي e، فإننا نكتب lnx. ويُطلق على الثابت الرياضي e اسم «عدد أويلر» ويساوي تقريبًا ٢٫٧١٨٢٨. في العلوم الطبيعية، نقابل الرمز lnx كثيرًا، وهذا يفسِّر تسميتها بـ «اللوغاريتم الطبيعي». أما الأساس المشترك الآخر فهو العدد ٢، وبدلًا من كتابة log_2x فإننا نكتب lgx. تكثر اللوغاريتمات ذات الأساس ٢ في علوم الكمبيوتر والخوارزميات، ولكن ربما لا تُستخدم في غير هذين المجالين، على الرغم من أننا تعرَّضنا لها بالفعل. في مثال طي الورقة، إذا كانت لفيفة ورق تحتوي على ١٠٢٤ طبقة، فقد طُوِيَت $lg1024 = lg2^{10} = 10$ مرات. وفي مثال لعبة الشطرنج، نتج عدد حبات الأرز عن عدد المضاعفات التي أجريناها، وهي $lg2^{63} = 63$.

والسبب في ظهور الرمز lgx كثيرًا في الخوارزميات هو أنه يظهر عندما نحل مسألة عن طريق تقسيمها إلى مسألتين فرعيتين متساويتين؛ وتسمَّى هذه الخوارزمية «فرِّق تسُد» وتشبه في عملها طي ورقة إلى طبقتين. وتُعد أفضل الطرق فاعلية وكفاءة للبحث عن شيء وسط مجموعة عناصر مرتَّبة تحتوي على التعقيد $O\ (lgn)$. هذا مذهل إلى حدٍّ ما؛ فهو يعني أنه لكي تعثر على شيء ضمن مليار عنصر مرتَّب، فلست بحاجة إلا إلى $lg10^9 \approx 30$ عملية استكشاف دقيق وسط العناصر.

تأتي الخوارزميات التي تحتوي على تعقيد لوغاريتمي في المرتبة الثانية من حيث الأفضلية بعد الخوارزميات التي تعمل ضمن وقت ثابت. وتليهما الخوارزميات ذات التعقيد $O\ (n)$، التي تسمَّى خوارزميات «الوقت الخطي»؛ لأن زمنها يزيد تناسبيًّا مع قيمة n؛ وهذا يعني أنها تزيد كمضاعفات لقيمة n. لقد رأينا أن البحث عن عنصرٍ في مجموعةٍ غير مرتبة من العناصر يتطلب وقتًا يتناسب مع عدد العناصر، $O\ (n)$. انظر كيف زاد التعقيد مقارنة بالوقت الذي كانت فيه العناصر مرتَّبة؛ فقد يكون لتنظيم البيانات في المسألة تأثيرٌ كبير في طريقة حلها. وبوجه عام، يعتبر الوقت الخطي هو السلوك الأفضل الذي يمكن أن نتوقَّعه من الخوارزمية إذا كانت مضطرة إلى قراءة كل مدخلات المسألة؛ إذ ستتطلب تلك العملية الوقت $O\ (n)$ من أجل العدد n من المدخلات.

إذا جمعنا الوقت الخطي والوقت اللوغاريتمي، فسنحصل على خوارزميات «الوقت اللوغاريتمي الخطي»، حيث يزداد وقت تلك الخوارزميات بمقدار حاصل ضرب قيمة n في اللوغاريتم الخاص بها، أي $nlgn$. وأفضل خوارزميات الفرز والترتيب — أي تنظيم العناصر — تحتوي على التعقيد $O\ (nlgn)$. قد يبدو هذا مفاجئًا بعض الشيء؛ وعلى كل

حال، ربما يتضح أنك إذا كان لديك العدد n من العناصر وأردت مقارنة كل عنصر مع كل العناصر الأخرى، فسيتطلب هذا المقدار $O\left(n^2\right)$ من الوقت، وهو أكبر من $O\left(nlgn\right)$.[9] إضافة إلى ذلك، إذا كان لديك العدد n من العناصر وتريد ترتيبها، فبالتأكيد تحتاج إلى مقدار الوقت $O\left(n\right)$ كي تفحص كل العناصر. إن ترتيب تلك العناصر يحتاج إلى ضرب ذلك العدد في عاملٍ أصغر من قيمة n نفسها. وسنعرف كيف يمكن تنفيذ ذلك في موضعٍ لاحق في الكتاب.

الفئة التالية من التعقيد الحسابي هي n مرفوعة إلى قوة أسية ثابتة، $O\left(n^k\right)$؛ وتسمَّى هذه الفئة «التعقيد المتعدد الحدود». وتُعد خوارزميات الوقت المتعدد الحدود خوارزمياتٍ فعَّالة إلا إذا كانت قيمة k كبيرة، ولكن نادرًا ما يحدث ذلك. عندما نحاول حل مسألة حسابية، فعادة ما نبتهج إذا توصَّلنا إلى خوارزمية ذات وقت متعدد الحدود.

يُطلق على التعقيد ذي الصيغة $O\left(k^n\right)$ اسم «التعقيد الأسي». لاحظ الفرق بينه وبين التعقيد المتعدد الحدود حيث كان الأس ثابتًا، أما هنا فالأس هو الذي يتغير. لقد رأينا السرعة الهائلة للنمو الأسي. لن يبقى الكون طويلًا بما يكفي كي يرى إجابة الخوارزميات الأسية للمدخلات غير التافهة. تلك الخوارزميات مثيرة للاهتمام من الناحية النظرية؛ لأنها تبيِّن أنه يمكن العثور على حل. يمكننا بعد ذلك البحث عن خوارزميات أفضل ذات تعقيد أقل، أو ربما نتمكَّن من إثبات أنه لا يوجد خوارزميات أفضل، وفي تلك الحالة يمكننا أن نركن إلى شيء لا يرتقي إلى المثالية، مثل الحلول التقريبية.

يوجد شيء يتنامى أسرع من القوة الأسية، وهذا الشيء هو «المضروب». إذا كنت لم تصادف المضروب من قبل، فهو عبارة عن عدد طبيعي n — نعبِّر عنه بالرمز $n!$ — ناتج ببساطة عن حاصل ضرب جميع الأعداد الطبيعية من ١ إلى ذلك العدد بما فيها العدد نفسه: ١٠٠! = ١ × ٢ × ٣ × ··· × ١٠٠. حتى لو لم تكن قد رأيت العدد ١٠٠! فربما رأيت العدد ٥٢! حتى من دون أن تعرفه. فهذا الرقم يعبِّر عن عمليات ترتيب عشوائي مختلفة لأوراق اللعب. وتحتوي الخوارزميات التي يُقاس وقت تشغيلها بالمضروب على «تعقيد مضروب».

على الرغم من أن الأعداد مثل ١٠٠! قد تبدو غريبة، فإنها تظهر في العديد من المواضع غير الغريبة وليس في ألعاب الورق فقط. لننظر إلى المسألة التالية على سبيل المثال: «إذا كانت لديك قائمة بعدة مدن والمسافات بين كل مدينتين فيها، فما أقصر طريق محتمل يجب أخذه لزيارة كل مدينة مرة واحدة ثم العودة إلى المدينة الأصلية؟» يُطلق على هذا المثال «مسألة البائع المتجول» والطريقة البديهية لحلها هي دراسة كل طريق محتمل

يمرُّ على جميع المدن. للأسف، للتعبير عن العدد n من المدن، سنكتب الرمز n! وسيتعذر التعامل مع المسألة بعد، لنقل، ٢٠ مدينة. توجد بعض الخوارزميات التي تحل المسألة بطريقةٍ أفضل بعض الشيء من $O(n!)$، ولكنها ليست عملية بالدرجة الكافية. قد يبدو ذلك مفاجئًا بالنسبة إلى مسألة مباشرة كتلك، ولكن الطريقة الوحيدة التي يمكن حلها بها خلال قدرٍ مقبول ومنطقي من الوقت هي إيجاد حلٍّ قد لا يكون الأمثل، ولكنه قريب إليه بدرجةٍ كافية. هناك العديد من المسائل الأخرى ذات الأهمية العملية الكبيرة التي «يستعصي حلها»، بمعنى أننا لا نعرف خوارزمية عملية لحلها حلًّا دقيقًا. على الرغم من ذلك، فالسعي وراء خوارزميات «تقريب» أفضل يُعد من المجالات النشطة في علوم الكمبيوتر.

في الجدول التالي، يمكن أن ترى قيمةَ دوالٍ متعددةٍ تندرج تحت فئات التعقيد التي تناولناها، تعبِّر عن قيمٍ مختلفة للرمز n. يوضح الصف الأول قيم n ويعبِّر أيضًا عن التعقيد الخطي؛ أما الصفوف التالية فتوضح فئات التعقيد المتصاعد. بما أن قيمة n تتصاعد، فهذا يعني تصاعد قيم الدوال، ولكن مع اختلاف الطرق التي تتصاعد بها. فتنقلنا الدالة n^3 من المليون إلى الكوينتليون، ولكنها لا تُقارن بأي حال بالدالة ٢^١٠٠ أو الدالة ١٠٠! وقد تركنا صفًّا فارغًا بعد الصف n^k للفصل بين الخوارزميات العملية وغير العملية. والحد الذي يفصل بين النوعين هو الخوارزميات المتعددة الحدود، التي تبيَّن أنها عملية في الاستخدام كما رأينا. أما الخوارزميات ذات التعقيد الأعلى، فعادة ما لا تكون عملية في الاستخدام.

n	1	10	100	1000	1000000
lgn	0	3.32	6.64	9.97	19.93
$nlgn$	0	33.22	664.39	9965.78	1.9×10^7
n^2	1	100	10000	1000000	10^{12}
n^3	1	1000	1000000	10^9	10^{18}
n^k	1	10^k	100^k	1000^k	1000000^k
2^n	2	1024	1.3×10^{30}	10^{301}	$10^{10^{5.5}}$
$n!$	1	3628800	9.33×10^{157}	4×10^{2567}	$10^{10^{6.7}}$

الفصل الثاني

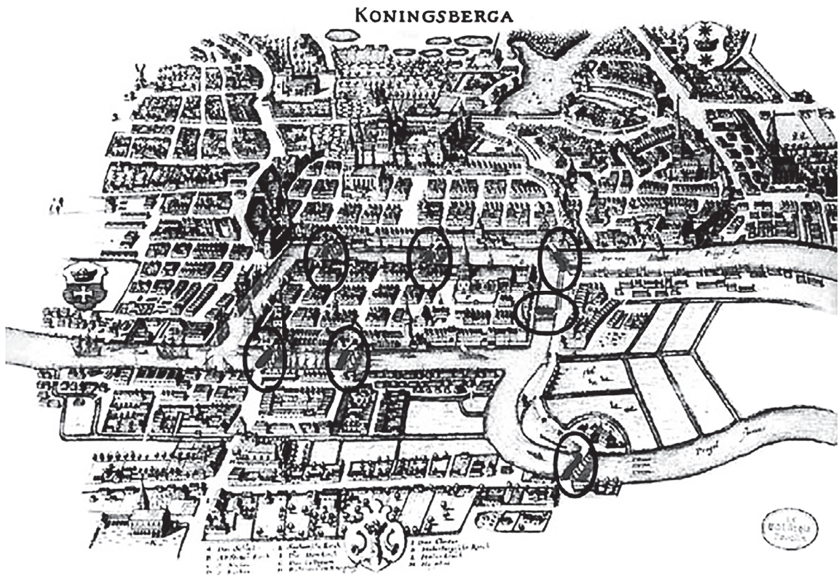
التمثيلات البيانية

في القرن الثامن عشر، كان مواطنو مدينة كونيجسبرج الشرفاء يتجولون في شوارع المدينة في أيام الأحد بعد الظهيرة. بُنيت مدينة كونيجسبرج على ضفاف نهر بريجل. تسبَّب النهر في تكوين جزيرتين كبيرتين داخل المدينة، وكانت الجزيرتان متصلتين بالبر الرئيسي وبإحداهما الأخرى من خلال سبعة جسور في المجمل.

اجتاحت مدينة كونيجسبرج تقلبات التاريخ الأوروبي، فانتقلت من سيطرة فرسان التيوتون ثم أصبحت عاصمة بروسيا واجتاحتها روسيا ثم جمهورية فايمر ثم ألمانيا النازية، وبعد الحرب العالمية الثانية، أصبحت جزءًا من الاتحاد السوفييتي وتغيَّر اسمها إلى كالينينجراد، وهو الاسم الحالي للمدينة. إنها جزء من روسيا اليوم على الرغم من أنها ليست متصلة بالأراضي الروسية. تقع كالينينجراد داخل جيب روسي على بحر البلطيق، بين كل من بولندا وليتوانيا.

في تلك الأيام، كانت المسألة التي تشغل عقول المواطنين الصالحين هي إذا ما كان بإمكانهم الانتهاء من جولاتهم بحيث لا يعبرون كل جسر من الجسور السبعة أكثر من مرة أم لا. ومِن ثَم سُمِّيت المسألة على اسم المدينة التي وقعت فيها لتصبح «مسألة جسور كونيجسبرج». وفيما يلي رسم لمدينة كونيجسبرج آنذاك للاطلاع على لمحةٍ عن طبيعة المسألة. ويُشار إلى الجسور في هذا الرسم بدوائرَ بيضاوية رُسِمت حولها. كانت المدينة تتكوَّن من جزيرتين، ولكنك لا ترى سوى جزيرة واحدة كاملة؛ لأن الجزيرة الأخرى تمتد إلى اليمين خارج حدود الخريطة.[1]

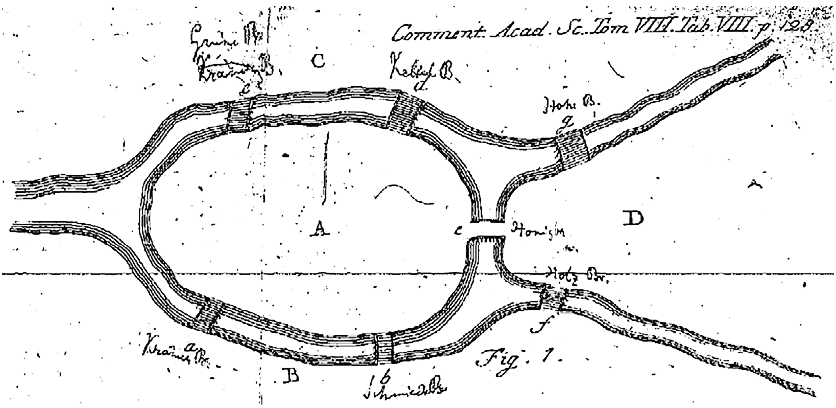
نمت المسألة إلى علم عالِم الرياضيات السويسري الشهير ليونهارت أويلر، لا نعرف تحديدًا كيف؛ إذ ورد ذكر المسألة في خطابٍ أُرسل يوم ٩ مارس ١٧٣٦ من عمدة مدينة



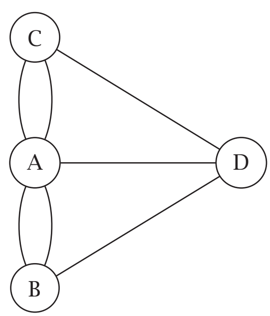
دانزيج التي تقع في بروسيا على مسافة ٨٠ ميلًا شرقي كونيجسبرج (تغيَّر اسم دانزيج إلى جدانسك وتتبع بولندا). يبدو أن مراسلة أويلر كانت جزءًا من جهود عمدة المدينة لتشجيع ازدهار الرياضيات في بروسيا.

كان أويلر آنذاك يعيش في مدينة سانت بطرسبرج في روسيا. فعمل على هذه المسألة وقدم حلًّا إلى أعضاء أكاديمية سانت بطرسبرج للعلوم يوم ٢٦ أغسطس ١٧٣٥. في العام التالي، كتب أويلر ورقة بحثية باللاتينية يذكر فيها الحل.[2] كان الحل «سلبيًّا»؛ إذ لم يكن ممكنًا القيام بجولة في المدينة بعبور كل جسر مرة واحدة فقط. قد تكون تلك الحقبة من تاريخ الرياضيات مثيرةً للاهتمام، ولكن بحل المسألة، أنشأ أويلر فرعًا جديدًا تمامًا في الرياضيات ألا وهو دراسة «التمثيلات البيانية».[3]

قبل أن نتناول التمثيلات البيانية، لنرى كيف تناول أويلر تلك المسألة. بادئ ذي بدء، جرَّد أويلر المسألة إلى عناصرها الأساسية. لا يلزم وجود خريطة مفصلة لمدينة كونيجسبرج لتمثيل المسألة. فقد رسم أويلر المخطط التالي لها:[4]



استخدم أويلر الحرفين A وD للإشارة إلى الجزيرتين، والحرفين B وC للإشارة إلى ضفتي البر الرئيسي. تمثَّلت الخطوة التالية في تجريد المخطط أكثر — بعيدًا عن الهندسة الفيزيائية — وتوضيح الروابط بين الجسور والجزيرتين والبر الرئيسي؛ لأن هذا ما يهم حقًّا فيما يتعلَّق بالمسألة:



رسمنا كتل اليابسة في شكل دوائر والجسور في شكل خطوط تصل بين تلك الدوائر. ومِن ثَم يمكن إعادة صياغة المسألة على النحو التالي: إذا كان معك قلم رصاص، فهل يمكنك البدء من أي دائرة، وتخفض القلم الرصاص على الورقة وتتبع الخطوط من دون رفع القلم من فوق الورقة بحيث لا يمكن المرور على كل خط أكثر من مرة؟

سار حل أويلر على النحو التالي. كلما دخلت إلى كتلة يابسة، لا بد أن تغادرها إلا لو كانت تلك الكتلة بداية جولتك أو نهايتها. وللقيام بذلك، لا بد أن يكون لكل كتلة يابسة — غير نقطتي البداية والنهاية — عدد زوجي من الجسور حتى يكون بإمكانك مغادرة تلك الكتلة من جسر مختلف في كل مرة تدخل إليها، كما هو مطلوب. انتقل الآن إلى الشكل وعُدَّ الجسور التي تربط كل كتلة يابسة. ستجد أن كل كتل اليابسة متصلة بعدد فردي من الجسور: الكتلة A لها خمسة جسور، بينما الكتل B وC وD لها ثلاثة جسور. وأيما كانت كتل اليابسة التي سنختارها لنقطتي البداية والنهاية، ستكون هناك كتل يابسة أخرى سنمر عليها في وسط الجولة، وكل كتلة من تلك الكتل لها عدد فردي من الجسور. ومِن ثَم لا يمكن الدخول إلى تلك الكتل والخروج منها دون عبور جسورها أكثر من مرة.

في الحقيقة، إذا وصلنا إلى النقطة B في أي وقتٍ من الجولة، فلا بد أننا عبرنا جسرًا كي نصل إليها. وسنعبر جسرًا آخر كي نغادرها. ولا بد أن نعبر الجسر الثالث في وقتٍ لاحق لأن المطلوب هو عبور كل الجسور. ولكن عندئذٍ سنعلق في الكتلة B نظرًا لعدم وجود جسر رابع، ولا يمكن أن نعبر جسرًا سبق أن عبرناه بالفعل. تسري المشكلة نفسها على الكتلتين C وD اللتين لهما ثلاثة جسور أيضًا. كذلك يسري الجدل نفسه على الكتلة A باعتبارها نقطةً وسيطة نظرًا لأن لها خمسة جسور؛ فبعد عبور الجسور الخمسة جميعًا للكتلة A، لن نتمكَّن من مغادرتها من جسرٍ سادس مختلِف نظرًا لعدم وجود هذا الجسر.

يتألَّف الشَّكل الذي رسمناه من دوائرَ وخطوطٍ تصل بين تلك الدوائر. وعلى سبيل استخدام المصطلحات الصحيحة، فقد أنشأنا بنيةً تتكوَّن من «عُقد» أو «رءوس» تتصل بـ «حواف» أو «روابط» تصل بينها. يطلق على البنية التي تتكوَّن من مجموعاتٍ من العُقَد والحواف «التمثيل البياني»، وكان أويلر أوَّل مَن عرَّف التمثيلات البيانية باعتبارها بنيةً، واستكشفَ خصائصَها. بلغة اليوم، تتعامل مسألة جسور كونيجسبرج السبعة مع «المسارات»، والمسار في التمثيل البياني هو سلسلة من الحواف تصل بين سلسلةٍ من العُقَد. إذن، فمسألة كونيجسبرج تدور حول إيجاد «مسار أويلري» أو «طريق أويلري»، وهو خطٌّ يمر من خلال تمثيل بياني بحيث يمر من كل حافة مرة واحدة فقط. والمسار الذي يبدأ وينتهي عند النقطة نفسها يسمَّى «جولة» أو «دورة». وإذا أضفنا أيضًا قيدًا (ليس في المسألة الأصلية) يتمثَّل في بدء مسار أويلري من نقطة وانتهائه عند النقطة نفسها، يصبح لدينا ما يطلَق عليه «جولة أويلرية» أو «دورة أويلرية».

تطبيقات التمثيلات البيانية كثيرة لدرجة أنها قد تملأ كتبًا كاملة. أي شيء يمكن تمثيله بعُقَدٍ متصلة بعُقَد أخرى يمكن توضيحه بتمثيل بياني. بمجرد أن نفعل ذلك،

يمكننا طرح كل الأسئلة المهمة بشأن التمثيل البياني؛ ولكن في هذا الكتاب، ستُتاح لنا الفرصة أن نلقي نظرة سريعة فقط.

ولكن قبل أن نفعله، سأقدِّم تفصيلة صغيرة لإرضاء القراء ذوي العقول الشديدة الدقة. لقد ذكرنا أن التمثيل البياني هو بنية تتكوَّن من مجموعة رءوس وحواف. في الرياضيات، لا تحتوي المجموعة على العنصر نفسه مرتين. ولكن في تمثيل مسألة كونيجسبرج، تظهر الحافة نفسها أكثر من مرة؛ على سبيل المثال، توجد حافتان بين النقطة A والنقطة B. تُميز الحافة بنقطة بدايتها ونقطة نهايتها؛ لذا فالحافتان بين A وB هما في الحقيقة مثيلان لحافة واحدة. وبناءً على ذلك، فإن مجموعة الحواف ليست مجموعة في الحقيقة؛ بل «مجموعة متعددة»؛ أي مجموعة تحتوي على عدة مثيلات لعناصرها. والتمثيل البياني لمسألة مدينة كونيجسبرج، بالمثل، ليس تمثيلًا بيانيًّا، بل «تمثيلًا بيانيًّا متعددًا».

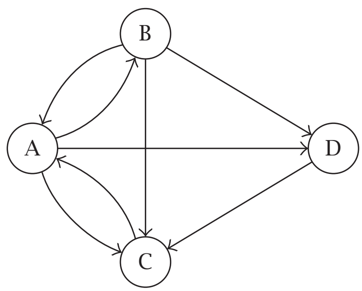
الانتقال من التمثيلات البيانية إلى الخوارزميات

إن تعريف التمثيل البياني شامل لدرجة أنه قد يضم أيَّ شيء يمكن تمثيله في صورة عناصر مرتبطة بعناصر أخرى. قد يتشابه التمثيل البياني في بعض الجوانب مع طبوغرافية مكانٍ ما، ولكن قد لا يكون للعُقَد والروابط علاقة بالنقاط في حيز المكان.

تُعَد «الشبكات الاجتماعية» مثالًا لمثل هذا التمثيل البياني. في الشبكات الاجتماعية، تمثِّل العُقَد الممثلين الاجتماعيين (وهؤلاء قد يكونون أفرادًا أو مؤسسات)، بينما تمثِّل الروابطُ التفاعلات بينهم. قد يكون الممثلون الاجتماعيون ممثلين حقيقيين، وقد تكون الروابط هي عمليات التعاون بينهم في الأفلام. قد نكون نحن الممثلين الاجتماعيين، والروابط قد تكون هي علاقاتنا مع الآخرين في أحد تطبيقات الشبكات الاجتماعية. عندئذٍ، يمكننا استخدام الشبكات الاجتماعية للعثور على جماعاتٍ من الناس، انطلاقًا من فرضية أن الجماعات تتكوَّن بفضل أشخاصٍ يتفاعلون بعضهم مع بعض. وتوجد خوارزميات تمتاز بالفاعلية في إيجاد جماعات في شكل رسوم بيانية بها ملايين العُقَد.

> إن تعريف التمثيل البياني شامل لدرجة أنه قد يضم أيَّ شيء يمكن تمثيله في صورة عناصر مرتبطة بعناصر أخرى.

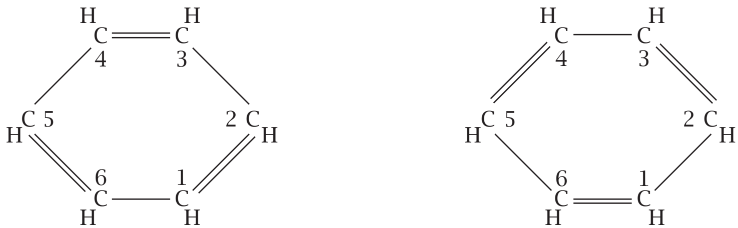
الحواف في التمثيل البياني لمسألة كونيجسبرج ليست موجهة، بمعنى أنه يمكن اجتيازها في الاتجاهين كليهما؛ على سبيل المثال، يمكننا الانتقال من النقطة A إلى النقطة B ومن النقطة B إلى النقطة A. ينطبق الأمر نفسه على الشبكات الاجتماعية حين تكون العلاقات متبادلة. وهذا ليس ضروريًّا على الدوام. فبناءً على تطبيقاتنا، يمكن أن تكون الحواف في تمثيل بيانيٍّ ما موجهة. وعندما تكون الحواف موجهة، نرسم الحواف مصحوبة بأسهم في أطرافها. يمكن رؤية تمثيل بياني موجَّه فيما يلي. مع ملاحظة أن هذا التمثيل ليس تمثيلًا بيانيًّا متعددًا؛ لأن الحافة من A إلى B ليست كالحافة من B إلى A.



وتُعَد الشبكة العنكبوتية العالمية مثالًا لتمثيل بياني موجَّه (ضخم). يمكننا تمثيل الشبكة بحيث ترمز العُقَد إلى صفحات الويب، بينما ترمز الحواف إلى الروابط التشعُّبية بين كل صفحتين. وهذا التمثيل البياني موجَّه؛ لأن الصفحة يمكن أن ترتبط بصفحةٍ أخرى، ولكن ليس بالضرورة أن تعود تلك الصفحة الثانية للارتباط بالأولى.

عندما يمكن البدء من عقدةٍ ما في التمثيل البياني، وعبور الحواف والعودة إلى العقدة التي بدأنا منها، نقول إن ذلك التمثيل البياني له «دورة». لكن ليست كل التمثيلات البيانية لها دورات. يحتوي التمثيل البياني في مسألة كونيجسبرج على دورات، على الرغم من أنه لا يحتوي على دورة أويلرية. ويعتبر التمثيل البياني الدوري الأشهر (وهو في الواقع تمثيل بياني متعدد) في تاريخ العلم نموذج أوجست كيكولي لبنية جزيئات البنزين:[5]

يُطلق على التمثيل البياني الذي ليس له دورة «تمثيل بياني غير دوري». تشكِّل التمثيلات البيانية غير الدورية الموجَّهة فئةً مهمة في التمثيلات البيانية. وللتمثيلات البيانية غير الدورية الموجَّهة استخدامات عديدة؛ على سبيل المثال، نستخدمها للتعبير عن الأولويات بين المهام (نعبِّر عن المهام بالعُقَد، والأولويات هي الروابط بينها)، وعلاقات الاعتماد على



الغير والمتطلبات الأساسية وغيرها من الترتيبات المشابهة. سننحي التمثيلات البيانية غير الدورية جانبًا الآن ونوجِّه انتباهنا إلى التمثيلات البيانية الدورية التي ستفتح لنا أول نافذة على الخوارزميات في التمثيلات البيانية.

المسارات والحمض النووي

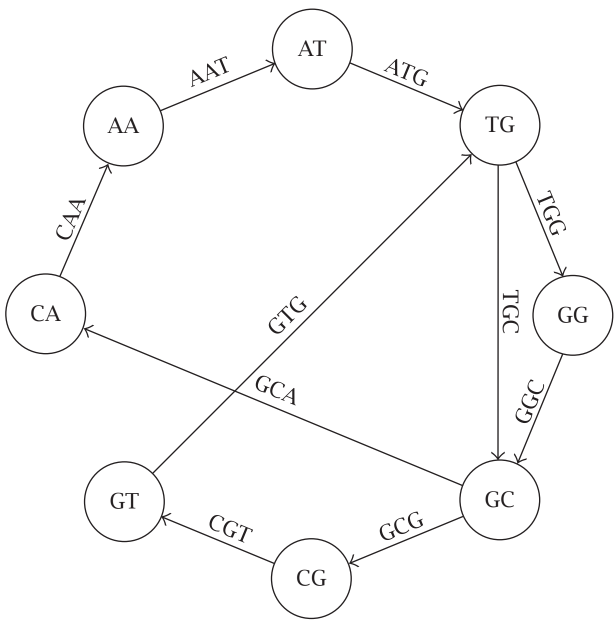
كان من أهم التطورات العلمية التي حدثت في العقود الأخيرة فكُّ شفرة الجينوم البشري. فبفضل التقنيات التي طُوِّرت في هذا الصدد، يمكننا الآن فحصُ الأمراض الجينية واكتشافُ الطفرات الجينية ودراسة جينومات الأنواع المنقرضة وغيرها من التطبيقات المذهلة.

يتم ترميز الجينوم في الحمض النووي، وهو عبارة عن جزيء عضوي كبير يتكوَّن من لولب مزدوج. يتكوَّن اللولب المزدوج من أربع قواعد هي: السيتوسين (C) والجوانين (G) والأدينين (A) والثايمين (T). يتكوَّن كل جزء في اللولب المزدوج من سلسلة من القواعد، مثل ACCGTATAG. أما الجزء الآخر من اللولب المزدوج فيتكوَّن من قواعد ترتبط بنظيراتها في الجزء الأول وفقًا للقاعدتين A–T وC–G. فإذا كان أحد جزأَي اللولب ACCGTATAG، فسيكون الجزء الثاني TGGCATATC.

لمعرفة تكوين حمض نووي غير معروف، يمكننا اتباعُ الإجراءات التالية. ننشئ نسخًا عديدة من السلسلة ونقسِّمها إلى أجزاءٍ صغيرة؛ كأن نقسمها إلى أجزاء يحتوي كلٌّ منها على ثلاث قواعد. وباستخدام أدوات متخصصة، يمكننا تحديد مثل هذه الأجزاء الصغيرة بسهولة. وبهذه الطريقة، نتوصَّل إلى مجموعة من الأجزاء المعروفة. بعد ذلك يتبقى أمامنا مشكلة تجميع الأجزاء في سلسلة حمض نووي، سنعرف تكوينها حينئذٍ.

لنفترض أن لدينا الأجزاء التالية، أو البوليمرات كما يسمونها: GTG وTGG وATG وGGC وGCG وCGT وGCA وTGC وCAA وAAT. يتكوَّن كل بوليمر من ثلاث قواعد؛ ولإيجاد سلسلة الحمض النووي التي تؤلِّفها تلك البوليمرات، ننشئ مخططًا بيانيًّا. في

هذا المخطط، تعبِّر الرءوس عن بوليمرات مكوَّنة من قاعدتين ومشتقة من بوليمرات مكوَّنة من ثلاث قواعد، ويأخذ من البوليمرات المكوَّنة من ثلاث قواعد أول بوليمرين وآخر بوليمرين. إذن، من البوليمر GTG، سنأخذ GT وTG، ومن البوليمر TGG، سنأخذ TG وGG. في المخطط البياني، نضيف حافة لكل واحد من البوليمرات الأولية أو المكوَّنة من ثلاث قواعد الذي استُخدم لاشتقاق الرأسين. نسمي تلك الحافة البوليمر. ومن البوليمر ATG نحصل على الرأسين AT وTG والحافة ATG. يمكنك الاطلاع على المخطط البياني الناتج عن هذا المثال:



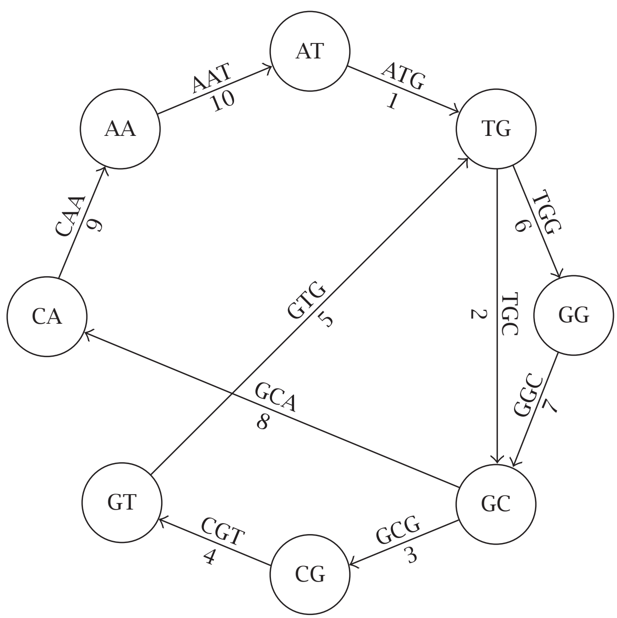
باستخدام المخطط البياني الذي أنشأناه، لن نحتاج إلا إلى إيجاد دورة في المخطط البياني تمر على كل الحواف مرة واحدة فقط — أي دورة أويلرية — من أجل إيجاد سلسلة الحمض النووي الأولية. وقد نُشرت الورقة البحثية «خوارزمية هيرهولزر» لإيجاد الدورات الأويلرية على المخططات البيانية على يد عالِم الرياضيات الألماني كارل هيرهولزر عام ١٨٧٣ وتسير كالتالي:[6]

(١) نحدِّد عقدة بداية.

(٢) ننتقل من عقدةٍ إلى أخرى حتى نعود إلى عقدة البدء. ليس بالضرورة أن تغطي الدورة التي تتبَّعناها جميع الحواف.

(٣) ما دام هناك رأس ينتمي إلى الدورة التي اتبعناها ولكنه في الوقت نفسه جزء من حافة ليست على المسار، نبدأ مسارًا جديدًا من ذلك الرأس باستخدام الحواف التي لم نستخدمها بعدُ حتى نعود إليها بحيث نكوِّن دورة أخرى. ثم نوصل تلك الدورة بالدورة التي اتبعناها بالفعل.

إذا استخدمنا الخوارزمية في التمثيل البياني المذكور في المثال، فسنحصل على مسارٍ بالشكل التالي:



لقد بدأنا من AT وقمنا بالدورة AT → TG → GG → GC → CA → AA → AT. لقد صنعنا دورةً ولكننا لم نغطِّ كلَّ الحواف. نرى أن TG لها حافة — TGC — لم نمرَّ بها بعدُ. لذا ننتقل إلى TG ونقوم بدورة تبدأ عبر الحافة TGC بحيث نحصل على المسار

$TG \rightarrow GC \rightarrow CG \rightarrow GT \rightarrow TG$. نضم المسار الثاني إلى المسار الأول، لنحصل بذلك على المسار الوارد في الشكل البياني: $AT \rightarrow TG\ (\rightarrow GC \rightarrow CG \rightarrow GT \rightarrow TG) \rightarrow GG \rightarrow GC \rightarrow CA \rightarrow AA \rightarrow AT$. إذا مشينا في المسار الناتج من العقدة الأولى إلى العقدة الأخيرة، دون لمس العقدة الأخيرة، وجمعنا الرءوس في سلسلةٍ واحدة بحيث نمر على القاعدة المشتركة مرة واحدة، فسنحصل على سلسلة الحمض النووي ATGCGTGGCA. يمكنك التحقُّق من أن تلك السلسلة تحتوي على جميع البوليمرات التي بدأنا بها؛ علمًا بأنك ستجد البوليمرات CAT وCAA إذا استدرت عندما تصل إلى نهاية السلسلة وعُدت إلى نقطة البداية.

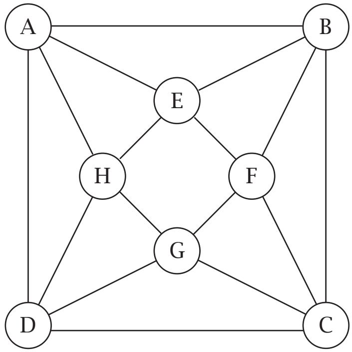
في هذا الشرح التوضيحي الخاص، أوجدنا مسارًا واحدًا إضافيًّا فقط وألحقناه بالمسار الأصلي. وبوجه عام، قد يكون هناك المزيد من المسارات؛ لأن الخطوة الثالثة من الخوارزمية تتكرَّر ما دام هناك رءوس ذات حواف لم نمرَّ بها بعد. تمتاز خوارزمية هيرهولزر بالسرعة؛ فإذا طُبقت تطبيقًا صحيحًا، فإنها تعمل في وقت خطِّي $O\ (n)$، حيث n ترمز إلى عدد الحواف في المخطط البياني.[7]

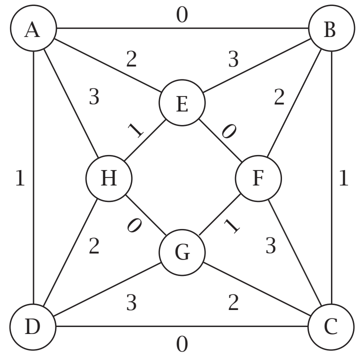
جدولة المسابقات الرياضية

لنفترض أنك تنظِّم بطولة رياضية سوف يتبارى فيها المتسابقون أزواجًا، ومِن ثَم سيكون لدينا سلسلة من المباريات. لدينا ثمانية متسابقين، وسيلعب كل متسابق أربع مباريات. تدور المسألة التي نحن بصددها حول طريقة جدولة المسابقة. فنحن نريد جدولة المباريات بحيث يلعب كل متسابق مباراة واحدة فقط في اليوم.

الحل البديهي هو أن نقيم مباراةً واحدة في اليوم ونترك المسابقة تستمر المدة التي تحتاج إليها. وبما أن لدينا ثمانية متسابقين وكل متسابق سيلعب أربع مباريات، فستستمر المسابقة ١٦ يومًا (٨ × ٤ / ٢؛ والقسمة على اثنين هنا بغرض عدم عدِّ كل مباراة مرتين). سنسمي المتسابقين الثمانية أليس (A) وبوب (B) وكارول (C) وديف (D) وإيف (E) وفرانك (F) وجريس (G) وهيدي (H). وهذا يتيح لنا استخدام الحرف الأول فقط من أسمائهم لتعريفهم.

يمكننا أن نجد حلًّا أفضلَ إذا أنشأنا مخططًا بيانيًّا لتمثيل المسألة. سنحدِّد رأسًا لكل لاعب وحافة لكل مباراة. عندئذٍ سيتخذ التمثيل البياني الشكل الموضَّح جهة اليمين أدناه. وعلى اليسار، ميزنا الحواف باسم اليوم الذي ستُقام فيه المباراة. كيف وجدنا ذلك الحل؟





نتَّفق على وضعِ أرقام أيام المباريات بترتيب متسلسل. لنقل إن البطولة ستبدأ في اليوم صفر. سنجدول كل المباريات، واحدة بواحدة.

(١) نأخذ مباراةً لم تُدرَج في الجدول بعد. علينا التوقُّف إذا كنَّا قد أدرجنا كل المباريات.

(٢) نحدِّد موعد المباراة في أقربِ يوم بحيث لا يشارك أيٌّ من اللاعبين في مباراة أخرى في ذلك اليوم. ثم نرجع إلى الخطوة الأولى.

تبدو هذه الخوارزمية بسيطةً إلى درجة خادعة، وقد يساورك الشك في قدرتها على حل المسألة حقًّا. لذا، دعنا نواصل ونرى ما يحدث. في الجدول التالي، يمكننا أن نرى المباريات، واحدة بواحدة، واليوم المحدَّد لإقامة كل مباراة، كما طبَّقنا الخوارزمية على التمثيل البياني. ينبغي قراءة أول عمودين في الجدول ثم العمودين التاليين:

المباراة	اليوم	المباراة	اليوم
A, B	0	C, F	3
A, D	1	C, G	2
A, E	2	D, G	3
A, H	3	D, H	2
B, C	1	E, F	0

المباراة	اليوم	المباراة	اليوم
B, E	3	E, H	1
B, F	2	F, G	1
C, D	0	G, H	0

نبدأ بمباراة أليس ضد بوب. لن يلعب أليس أو بوب أيَّ مباراة أخرى في اليوم صفر؛ أي في اليوم الذي سنحدِّده لإقامة المباراة.

بعد ذلك نأخذ مباراةً أخرى لم تُدرج في الجدول بعد، ولننقل مباراة أليس ضد ديف. سنرتِّب أسماء اللاعبين ترتيبًا أبجديًّا، على الرغم من عدم ضرورة ذلك في مسيرتنا، ولكن تذكَّر أنه كان يمكننا إدراجهم بترتيبٍ آخر — حتى ولو كان عشوائيًّا — ما دمنا نتعامل مع كل مباراة مرة واحدة فقط. لدى أليس مباراة أُدرجت في اليوم صفر بالفعل؛ لذا فإن أول يوم متاح للمباراة هو اليوم واحد.

يأتي بعد ذلك المباراة بين أليس وديف. أُدرجت أليس في اليوم صفر واليوم واحد، ومِن ثَم سندرج المباراة في اليوم اثنين. آخر مباراة لأليس ستكون مع هيدي؛ فأليس مشغولة في الأيام صفر وواحد واثنين ومن ثم سندرج هذه المباراة في اليوم ثلاثة.

انتهينا من أليس. ننتقل إلى مباريات بوب؛ باستثناء اليوم الذي أدرجناه كي يلعب فيه ضد أليس، سنحتاج إلى وضع خطة لمباراة بوب ضد كارول. أُدرِج بوب بالفعل في اليوم صفر (مع أليس)؛ ومِن ثَم سيكون لزامًا أن تُقام هذه المباراة في اليوم واحد. عندما ندرج بوب مع إيف، نلاحظ أن بوب مشغول بالفعل في اليوم صفر واليوم واحد (أدرجنا هاتين المباراتين لتونا)، بينما أُدرِج إيف للعب ضد أليس في اليوم اثنين؛ ولذا أدرجنا مباراة بوب ضد إيف في اليوم ثلاثة. بالانتقال إلى مباراة بوب ضد فرانك، نجد أن بوب لديه مباريات في اليوم صفر واليوم واحد، ولكنَّه متفرغ في اليوم اثنين بينما فرانك ليس لديه مباريات على الإطلاق حتى ذلك اليوم. إذن، نقيم مباراة بوب ضد فرانك في اليوم «اثنين»، قبل مباراة بوب ضد إيف.

بعد بوب، سنلتفت إلى مباريات كارول. لم تُدرَج كارول ولا ديف في مباريات اليوم صفر، ومِن ثَم ستُقام مباراة كارول ضد ديف في اليوم الأول من البطولة. بعد ذلك،

يمكن إقامة مباراة كارول ضد فرانك في اليوم ثلاثة؛ لأن كارول لديها مباراتان في اليوم صفر (وهي تلك التي رتَّبنا لها لتونا) واليوم واحد (مع بوب كما رتَّبنا من قبل)، بينما يلعب فرانك ضد بوب في اليوم اثنين (كما رتَّبنا من قبل أيضًا). ستُقام مباراة كارول ضد جريس «قبل ذلك» في اليوم اثنين؛ لأن جريس ليس لديها مباريات أخرى مزمعة بعدُ، وكارول لا تزال متفرغة في اليوم اثنين.

نواصل على هذا المنوال مع باقي المباريات؛ المثير في الأمر أن المباريات في المربَّعات الداخلية والخارجية من التمثيل البياني ستُقام في أول يومين. تلك المباريات عبارة عن مجموعتين مختلفتين ستُلعبان بالتوازي قبل أن يبدأ اللعب بينهما. في النهاية، كان في الحل الذي وجدناه تحسنٌ كبير على الحل الساذج الذي يتطلب ١٦ يومًا؛ إذ لن نحتاج إلى أكثر من أربعة أيام!

تُعَد مسألة جدولة المسابقات تلك، في الحقيقة، مثالًا لمسألةٍ أعم، ألا وهي مسألة «تلوين الحواف». وتلوين الحواف في التمثيل البياني هو تعيين ألوان للحواف بحيث لا يكون لحافتين متلاصقتين لونٌ واحد. وفي هذا المثال، سنتعامل مع الألوان على سبيل المجاز. في المثال المذكور، تُعبِّر الألوان عن الأيام؛ وبوجه عام، يمكن أن تَكُون أي مجموعة أخرى من القيم المختلفة. وإذا أردنا تلوين الرءوس في المخطَّط البياني بدلًا من الحواف، بحيث لا يتشارك رأسان تربطهما حافة واحدة لونًا واحدًا، يصبح بين يدينا مسألة «تلوين الرءوس». ولا عجب في أن كلًّا من تلوين الحواف وتلوين الرءوس ينتمي إلى فئةٍ أكبر وهي مسائل «تلوين التمثيل البياني».

تُعُد خوارزمية تلوين الحواف التي تحدَّثنا عنها بسيطةً وفعَّالة (فهي تتعامل مع كل حافة على حدة، ومرة واحدة فقط). وتُعرف باسم «الخوارزمية الجشعة». والخوارزميات الجشعة هي خوارزميات تحاول حل المسألة عن طريق إيجاد أفضل الحلول «عند كل مرحلة» وليس الحل المثالي للمسألة ككل. تفيد الخوارزميات الجشعة في العديد من المسائل عندما يكون علينا تحديد خيار عند كل مرحلة من الحل وتكون القاعدة الحاكمة لنا هي «إيجاد الحل الأفضل في الوقت الحالي». والاستراتيجيات التي توجِّه اختياراتنا في تقييم خوارزميةٍ ما يُطلق عليها «الاستدلال» (heuristics) والكلمة مأخوذة من اللفظة اليونانية heuriskein التي تعني «الإيجاد» (أي الحل).

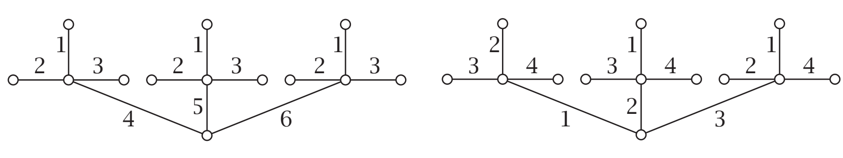
يمكننا أن ندرك ببعض التفكير أن ما يبدو أنه الأفضل في الوقت الحالي في الخوارزميات — وكذلك في الحياة الواقعية — قد لا يكون الاستراتيجية المثلى. قد يكون من المفيد أن نؤجِّل الشعور بالرضا؛ فالخيار الأفضل الآن ربما يقودنا إلى فخٍّ نندم عليه لاحقًا.

تخيَّل أنك تتسلَّق جبلًا. قد يدفعك الاستدلال الجشع إلى اختيار أشد المسارات انحدارًا عند كل نقطة (على افتراض أنك تتمتَّع ببراعةٍ لا تُضاهى في التسلُّق). ليس بالضرورة أن يأخذك هذا المسار إلى القمة، فقد يقودك إلى مرتفعٍ ليس عنده طريق سوى طريق العودة. قد يكون الطريق الحقيقي للوصول إلى القمة بين المنحدرات الأخف انحدارًا.

تُستخدم استعارة التسلُّق كثيرًا في حل المسائل في علوم الكمبيوتر. فنقوم بتمثيل المسألة التي بين يدينا بحيث يكمُن الحل عند «قمة» الخطوات التي يمكن أن نقوم بها ونحاول العثور على الخطوات الصحيحة؛ فيما يُسمى ﺑ «أسلوب تسلُّق التل». عندما نصل إلى شيءٍ أشبه بمرتفع، نقول إننا وصلنا إلى «الحل الأمثل الموضعي» ولكن ليس إلى «الحل الأمثل الشامل» وهو أعلى قمَّة نسعى للوصول إليها.

عودة من صعود التل إلى جدولة المسابقات، لقد اخترنا أول يوم متاح لكل مباراة. للأسف، قد لا تكون هذه الطريقة المثلى لجدولة كل المباريات. والواقع أنه قد تبيَّن بالفعل أن تلوين المخطَّط البياني مسألة صعبة. والخوارزمية التي تناولناها «ليس» مضمونًا أن تقدِّم الحل الأمثل؛ بمعنى أن الحل يتطلب أصغر عدد من الأيام (أو الألوان بوجه عام). يُطلق على عدد الحواف المتاخمة لعقدةٍ ما «درجة العقدة». ويمكن إثبات أنه إذا كانت أكبر درجة لأي عقدة في التمثيل البياني تساوي d، فإنه يمكن تلوين الحواف بعدد d أو $d + 1$ من الألوان على الأكثر؛ ويطلق على عدد الألوان اللازمة لحواف التمثيل البياني «الدليل اللوني» للمخطط البياني. في المثال الذي بين أيدينا، الحل مثالي، حيث $d = 4$، ونحن استخدمنا أربعة أيام. ولكن قد لا تتمكَّن الخوارزمية من إيجاد الحل المثالي في تمثيلٍ بيانيٍّ آخَر. بل قد تعطينا حلًّا أسوأ من ذلك. الشيء الجيد في تلوين التمثيل البياني الجشع هو أننا نعرف إلى أي مدًى قد يكون هذا الحل بعيدًا؛ بمعنى أن الحل الذي تقدِّمه قد يحتاج حتى $2d - 1$ من الألوان بدلًا من d، ولكن لن يكون الحل أسوأ من ذلك.

إذا كنت تريد أن تعرف كيف يمكن أن يحدث هذا، ففكر في تمثيلٍ بيانيٍّ يتكوَّن من «نجوم» متصلة بعقدة مركزية، كما في التمثيل البياني التالي:



إذا كان لدينا العدد k من النجوم، حيث كل نجم له العدد k من الحواف بالإضافة إلى حافة للعقدة المركزية، وبدأنا بتلوين النجوم، فسنستخدم العدد k من الألوان كي نلوِّن حواف النجوم. إذن، سنحتاج إلى عدد k إضافي من الألوان كي نصل النجومَ بالعقدة المركزية. فيكون إجمالي عدد $2k$ من الألوان. هذا ما فعلناه جهة اليسار. ولكن هذا ليس الحل الأمثل. إذا بدأنا بتلوين الحواف التي تصل النجوم بالعقدة المركزية، فسنحتاج إلى العدد k من الألوان. عندئذٍ يمكننا تلوين النجوم نفسها باستخدام لون واحد إضافي فقط، وبذلك يصبح الإجمالي $k + 1$ من الألوان. يمكننا الاطلاع على طريقة هذا الحل من الرسم جهة اليمين. كل هذا يتوافق مع النظرية؛ إذ إن كل نجم له الدرجة $k + 1$.

المشكلة أن الخوارزمية الجشعة تقرِّر ترتيب الحواف للتلوين بطريقةٍ ليست مثالية في النهاية، أو تحريًّا للمصطلحات السليمة، بطريقةٍ ليست «مثالية في العموم». ربما تتوصَّل إلى الحل الأفضل، ولكن قد لا يكون هو الأفضل. لذا نكرِّر مرة أخرى، أن الاختلاف عن الحل الأمثل ليس كبيرًا للدرجة. ولعل في هذا مدعاة للارتياح؛ لأن تلوين التمثيل البياني بالغ الصعوبة لدرجة أننا لو أردنا خوارزمية دقيقة يمكنها إيجاد أفضل حل لكل تمثيل بياني، فستحتوي الخوارزمية على تعقيدٍ أُسيٍّ بقيمة $O\left(2^n\right)$ تقريبًا، حيث n تساوي عدد الحواف في التمثيل البياني. ولذا فلا فائدة تُرجى من خوارزميات تلوين الحواف، إلا في التمثيلات البيانية الصغيرة.

تتضمَّن الخوارزمية الجشعة التي تناولناها خاصيةً أخرى جميلة (فضلًا عن كونها عملية). فهي تتسم بكونها «خوارزمية فورية»؛ بمعنى أنها خوارزمية تصلح حتى لو كانت المدخلات غير معروفة عند البدء، بل تظهر أثناء سير خطوات الحل. فلسنا بحاجة إلى معرفة كل الحواف كي نبدأ في استخدام الخوارزمية. ستعمل الخوارزمية بطريقةٍ صحيحة حتى إن أُنشِئ المخطط البياني تدريجيًّا — أي ببناء حافة واحدة في المرة الواحدة — بينما نقوم بتشغيل الخوارزمية. سوف يحدث هذا إذا كان اللاعبون يشتركون في المسابقة حتى بعد البدء في جدولة المباريات. سنتمكَّن من تلوين كل حافة (مباراة) تدخل إلى التمثيل البياني، وعندما ينتهي التمثيل البياني، سيكون لدينا تلوين جاهز للحواف. إضافة إلى ذلك، تُعَد هذه الخوارزمية الجشعة هي الخوارزمية المثلى إذا أُنشئ التمثيل البياني تدريجيًّا على هذا النحو؛ فلا توجد خوارزمية دقيقة على الإطلاق — مهما كانت عديمة الكفاءة — عند إنشاء التمثيل البياني في أثناء حل المسألة.[8]

أقصر المسارات

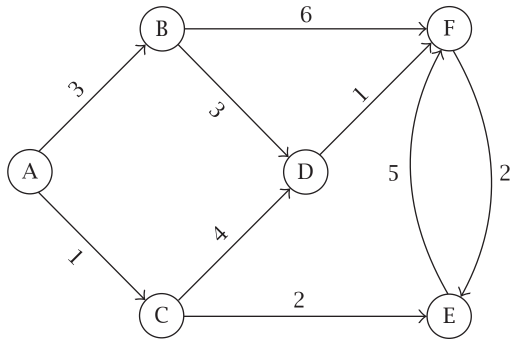
كما رأينا، تعمل الخوارزمية الجشعة باتخاذ القرار الأفضل عند كل خطوة، وربما لا يكون القرار الأفضل على الإطلاق. تتسم الخوارزمية بطبيعة انتهازية نوعًا ما أو طابع من اغتنام الفرص. وللأسف، وكما تخبرنا حكاية يعسوب، قد يندم الجندب الذي يعيش اليوم بيومه دون تحسُّب للمستقبل حينما يحل الشتاء، بينما النمل الذي يستعد للمستقبل، ينتهي به المطاف في راحة ودفء.[9] في التخطيط للمسابقات، وجدنا أن نهاية الجندب قد لا تكون بذلك السوء. والآن، حان وقت انتقام النمل.

في الفصل الأول، تناولنا عدم جدوى محاولة إيجاد أقصر المسارات بين نقطتين على شبكة عن طريق عدِّ كل المسارات المحتملة. ورأينا أن القيام بذلك مستحيل عمليًّا؛ لأن عدد المسارات يتزايد على نحوٍ هائل. أما وقد تعرَّفنا على التمثيلات البيانية الآن، فسنرى أن هناك طريقة للقيام بذلك. في الحقيقة، سنرتقي بالمسألة إلى مستوًى أعلى بعض الشيء. فبدلًا من البحث عن أقصر المسارات على شبكةٍ ما، لها شكل هندسي جميل وتتساوى فيها المسافات بين النقاط، سنسمح بأي شكل هندسي بل سنضيف مسافات مختلفة بين النقاط.

للقيام بذلك، سننشئ مخططًا بيانيًّا يحتوي على حوافَّ وعُقَد تمثِّل خريطة، ونريد إيجاد أقصر مسار بين عقدتين على الخريطة. إضافة إلى ذلك، سنلحق «وزنًا» بكل حافة. قد يكون الوزن موجبًا أو صفرًا، وسنربطه بقياس للمسافة بين عقدتين متصلتين. قد يكون هذا القياس مسافةً بالأميال أو زمنَ سيرٍ بالساعات؛ أيُّ وحدة قياس غير سالبة ستفي بالغرض. «طول المسار» يساوي مجموع الأوزان على طول المسار؛ ومِن ثَم يكون «أقصر مسار» بين عقدتين هو المسار الأقل طولًا. إذا كانت جميع الأوزان تساوي واحدًا، فإن طول المسار يساوي عدد الحواف على ذلك المسار. بمجرد أن نعطي الأوزان قيمًا أخرى، لن تصبح تلك القيمة صحيحة بعد الآن.

في المخطَّط البياني التالي، لدينا ست عُقَد متصلة بتسع حواف ذات أوزان متفاوتة، ونريد إيجاد أقصر المسارات للانتقال من العقدة A إلى العقدة F.

إذا اتبعنا الاستدلال الجشع، فسنبدأ من العقدة A إلى العقدة C؛ وهكذا يصبح أفضل خيار هو الانتقال إلى العقدة E، ومنها نسلك طريقنا إلى العقدة F. إجمالي طول المسار A وC وE وF يساوي ثمانية، ولكنه ليس المسار الأفضل. أفضل مسار هو الانتقال من العقدة A إلى العقدة C ثم إلى العقدة D، وبعدها إلى العقدة F بإجمالي طولٍ يساوي ستة.



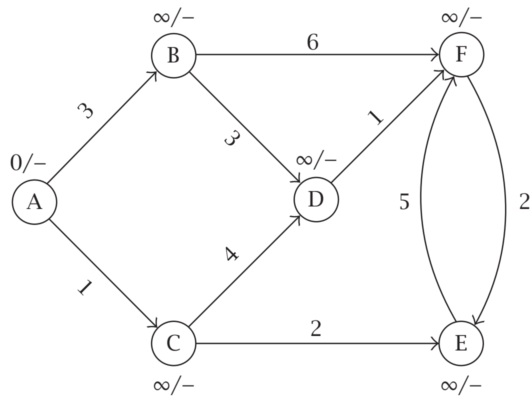
إذن فالاستدلال الجشع لا يصلح، وعلى العكس من تخطيط المسابقات، لا توجد ضمانات بشأن أسوأ أداء له فيما يتعلَّق بأقصر مسارٍ فعلي. على الرغم من ذلك، وخلافًا لتخطيط المسابقات مرة أخرى، توجد خوارزميات فعَّالة في إيجاد أقصر المسارات؛ ومِن ثَم فلا داعي لاستخدام الاستدلال الجشع في الواقع على الإطلاق.

في عام ١٩٥٦، كان هناك عالِم كمبيوتر هولندي شاب، يُدعى إيدسكر ديكسترا يتسوَّق في أمستردام مع خطيبته. لما تعبا، جلسا في مقهًى لاحتساء كوبين من القهوة، حيث أخذ ديكسترا يفكِّر في مسألة إيجاد أفضل الطرق للانتقال من مدينة إلى أخرى. وصمَّم الحل في غضون ٢٠ دقيقة على الرغم من أن الخوارزمية استغرقت بعض الوقت — ثلاث سنوات — حتى نشرها. حظيَ ديكسترا بمسيرةٍ مهنية لامعة، لكن ما أثار دهشته أن هذا الاختراع الذي استغرق ٢٠ دقيقة ظل حجرَ الأساس لشهرته.[10]

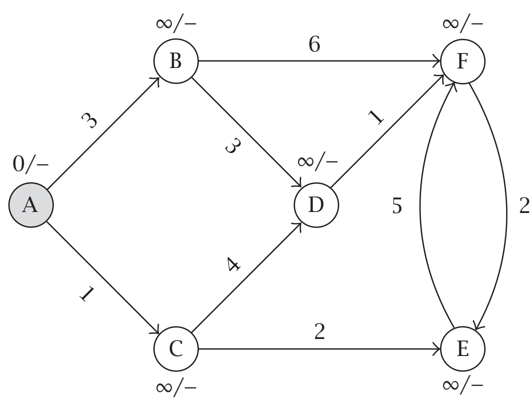
إذن، ما هي خطوات الخوارزمية؟ نريد إيجاد أقصر المسارات من عقدة إلى كل العُقَد الأخرى في مخطَّط بياني. تَستخدم الخوارزمية فكرةً يُطلق عليها «التخفيف»، وفيها نعيِّن تقديراتٍ للقيم التي نريد إيجادها (المسافات في هذه المسألة). في البداية، تكون التقديرات هي الاحتمال الأسوأ. ثم مع التقدُّم في خطوات الخوارزمية، نتمكَّن من تخفيف تلك التقديرات وتحويلها من التقديرات الشديدة السوء التي بدأنا بها إلى تقديراتٍ أفضلَ وأفضلَ تدريجيًّا حتى نصل إلى القيم الصحيحة.

في خوارزمية ديكسترا، يسير التخفيف على النحو التالي. نبدأ بتعيين أسوأ القيم المحتملة للمسافات من عقدة البداية إلى جميع العُقَد، فنعيِّن المسافة إلى ما لانهاية؛ ولا يخفى أنه لا يوجد ما هو أسوأ من ذلك! في الشكل التالي، وضعنا التقدير الأولي لأقصر مسار والعقدة السابقة في ذلك المسار، سواء فوق كل عقدة أو تحتها. بالنسبة إلى العقدة

A، لدينا القيمة −/0 لأن المسافة من A إلى A تساوي صفرًا ولا توجد عقدة سابقة على A. أما بالنسبة إلى جميع العُقَد الأخرى، لدينا القيمة −/∞؛ لأن المسافة لا نهائية وليس لدينا فكرة عن أقصر مسار إليها.

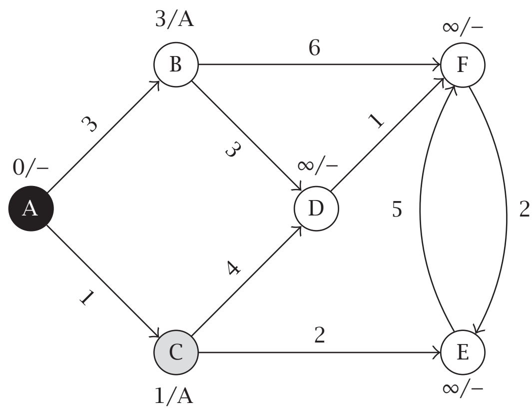


نأخذ العقدة الأقصر مسافة من A حتى الآن. إنها العقدة A نفسها. إنها العقدة الحالية، ومِن ثَم نظلِّلها باللون الرمادي.



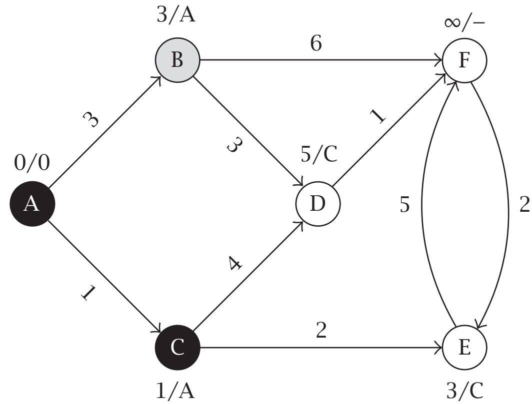
من العقدة A، يمكننا التحقُّق من التقديرات لأقصر المسارات إلى العُقَد المجاورة لها وهي العقدة B والعقدة C. في البداية، عينَّا هذه التقديرات إلى ما لانهاية، ولكننا نجد الآن

أنه يمكننا في الواقع الانتقال من B إلى A بقيمة تساوي ٣، ومن العقدة C إلى العقدة A بقيمة تساوي ١. نحدِّث هذه التقديرات ونشير أيضًا إلى أن التقديرات تمرُّ من على العقدة A؛ ومِن ثَم نكتب 3/A فوق العقدة B و1/A أسفل العقدة C. وبذلك نكون قد انتهينا من العقدة A لما تبقى من الخوارزمية. نحدِّث الشكل بناءً على ذلك ونظلل العقدة A باللون الأسود. ننتقل إلى عقدةٍ لم نمرَّ عليها وتتضمَّن أفضل تقدير في الوقت الحالي. إنها العقدة C.

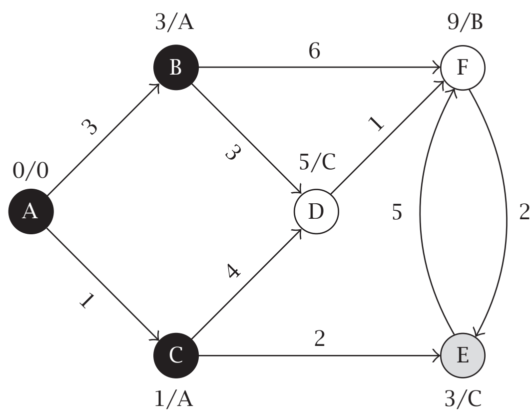


من العقدة C، نتحقَّق من تقديرات أقصر المسارات إلى العُقَد المجاورة لها وهي العقدة D والعقدة E. كانت التقديرات لا نهائية، ولكننا نرى الآن أنه يمكننا الوصول إلى كل واحدة منها من خلال العقدة C. المسار من A إلى D مرورًا بالعقدة C يساوي ٥ إجمالًا، ومِن ثَم نكتب 5/C فوق العقدة D. المسار من A إلى E مرورًا بالعقدة C يساوي ٣ إجمالًا، ومِن ثَم نكتب 3/C أسفل العقدة E. انتهينا من العقدة C ولذا نظللها باللون الأسود وننتقل إلى عقدةٍ لم نمرَّ عليها وتتضمَّن أفضل تقدير حتى الآن. تتضمَّن العقدتان B وE تقديرات جيدة على السواء بقيمة ٣. يمكننا اختيار أيٍّ منهما. لنختر العقدة B.

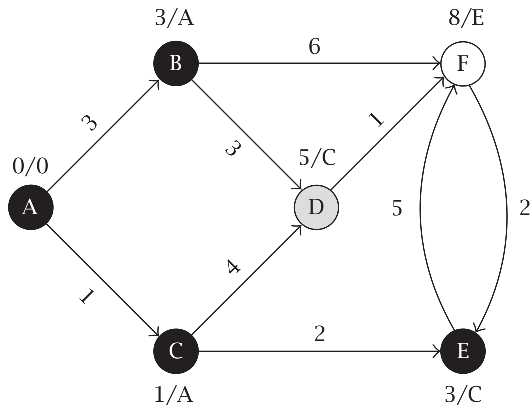
نعمل بالطريقة نفسها. من العقدة B، نتحقَّق من تقديرات أقصر المسارات إلى العُقَد المجاورة لها وهي العقدة D والعقدة F. لدينا بالفعل تقدير بطول ٥ للعقدة D بداية من العقدة C؛ هذا أفضل من الطول ٦ الذي كنا سنحصل عليه لو بدأنا من عند العقدة B. ومِن ثَم نبقي تقدير المسافة إلى العقدة D من دون تغيير. التقدير الحالي للعقدة F لا نهائي ومِن ثَم سنحدِّثه إلى ٩ مبتدئين من العقدة B. نظلل العقدة B للدلالة على المرور



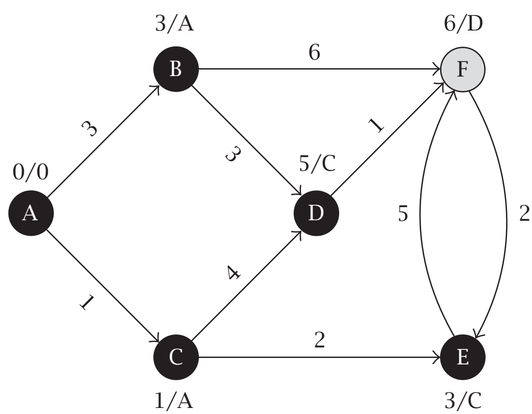
بها وننتقل إلى العقدة التي لم يتم المرور بها وتتضمَّن أفضل تقدير حتى الآن. إنها العقدة E.



العقدة E تجاورها العقدة F. طول المسار إلى العقدة F من العقدة E يساوي ٨، ما يجعله أفضل من المسار الذي أوجدناه من العقدة B. نحدِّث المسار ونظلِّل العقدة E دلالة على المرور بها وننتقل إلى العقدة التي لم يتم المرور بها ولها أفضل تقدير حتى الآن، وهي العقدة D.

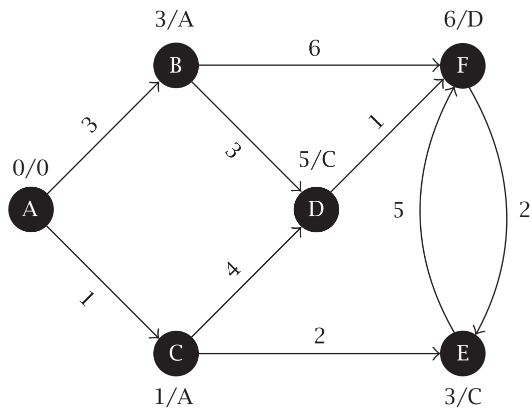


العقدة D تجاورها العقدة F، حيث طول المسار من النقطة E يساوي ٨. بما أنه يمكننا الوصول إلى F مرورًا بالعقدة D بطول يساوي ٦ إجمالًا، نحدِّث ذلك المسار. وكما في السابق، نمر علها وتتضمن أفضل تقدير حتى الآن، العقدة التي لم يتم المرور عليها حتى الآن هي العقدة F.



من العقدة F، نتحقَّق مما إن كان ينبغي أن نُحدِّث تقديرنا للعقدة E المجاورة لها أم لا. المسار الحالي إلى العقدة E يساوي ٣، بينما المسار عبر العقدة F ستبلغ قيمته ١٠.

نبقي العقدة E من دون تغيير. المرور على العقدة F لم يحدِث أيَّ فرق، ولكننا لم نستطِع معرفة هذا من قبل. بما أننا قد مررنا على جميع العُقَد، تنتهي الخوارزمية.



بينما كنَّا نسير بين خطوات الخوارزمية، كنا نسجِّل أطوال المسارات والعُقَد السابقة على كل عقدة عبر أقصر مسار. وقد فعلنا ذلك بحيث إذا أردنا — بعد الانتهاء من الخوارزمية — إيجاد أقصر مسار من العقدة A إلى أي عقدة أخرى في التمثيل البياني، ولنقل العقدة F، نبدأ من النهاية ونسلك طريقنا إلى عقدة البداية. نطلع على العقدة السابقة لها، وهي العقدة D. ونتوصَّل إلى العقدة السابقة على D ألا وهي العقدة C، والعقدة السابقة على C وهي العقدة A. المسار الأقصر من A إلى F هو A وC وD وF بإجمالي طول يساوي ستة، كما ذكرنا من قبل في بداية نقاشنا.

في النهاية، توصَّلت خوارزمية ديكسترا إلى «كل المسارات الأقصر» من عقدة البداية إلى كل العُقَد الأخرى في التمثيل البياني. الخوارزمية فعَّالة؛ إذ تساوي درجة تعقيدها $O((m+n)\,logn)$، حيث m تعبِّر عن عدد الحواف في التمثيل البياني، وn تعبِّر عن عدد العُقَد. وفيما يلي الخوارزمية في شكل خطوات:

(١) تعيين مسافةٍ لما لانهاية لجميع العُقَد باستثناء عقدة البداية؛ وتعيين مسافة تساوي صفرًا لعقدة البداية.

(٢) إيجاد العقدة التي لم يتم المرور بها ذات المسافة الأقل. ستكون هذه هي العقدة الحالية. إذا لم يتبقَّ عُقَد لم يتم المرور بها، توقف.

(٣) التحقُّق من جميع العُقَد المجاورة للعقدة الحالية. إذا كانت المسافات إلى تلك العُقَد أكبرَ من المسافة التي سنحصل عليها بدايةً من العقدة الحالية وقبل الوصول إلى العُقَد المجاورة لها، نخفِّف المسافة ونحدِّث المسار إلى العقدة المجاورة. ننتقل إلى الخطوة ٢.

إذا كان اهتمامنا منصبًّا فقط على إيجاد أقصر مسار إلى عقدةٍ محدَّدة، يمكننا التوقُّف عندما نختار المرور بها في الخطوة ٢. وبمجرد الانتهاء من ذلك، نكون قد وجدنا أقصر مسار إلى تلك العقدة، ولن يتغير فيما تبقى من خطوات تنفيذ الخوارزمية.

يمكن استخدام خوارزمية ديكسترا في أي تمثيل بياني حتى إذا كان يتضمَّن دورات، بشرط ألَّا يحتوي على أوزان سالبة. قد تحدث الأوزان السالبة إذا كانت الحواف تمثل نوعًا من المكافآت والعقوبات بين العُقَد. الخبر السار أنه توجد خوارزميات أخرى فعالة يمكننا استخدامها في حالة وجود أوزان سالبة، ولكن هذا يبيِّن أنه قد يكون لتلك الخوارزميات متطلبات خاصة في تطبيقها. وعندما نحاول إيجاد خوارزمية لحل مسألةٍ ما، ينبغي أن نتأكد من أن المسألة تستوفي متطلبات تلك الخوارزمية. وإلَّا فلن تؤتي الخوارزمية أيَّ فائدة؛ ولكن اعلم أن الخوارزمية لا تخبرنا بأنها عديمة الجدوى. إذا نفَّذنا الخوارزمية على جهاز كمبيوتر، فسيظل الجهاز ينفِّذ خطواتها حتى لو لم يكن لذلك أي فائدة. ومِن ثَم ستؤدي إلى إجابةٍ بلا أي معنًى. فالأمر يعود إلينا نحن في التأكُّد من أننا نستخدم الأداة المناسبة للمهمة المناسبة.

عندما نحاول إيجاد خوارزمية لحل مسألةٍ ما، ينبغي أن نتأكد من أن المسألة تستوفي متطلبات تلك الخوارزمية. وإلا فلن تُؤتي الخوارزمية أيَّ فائدة؛ ولكن اعلم أن الخوارزمية لا تخبرنا بأنها عديمة الجدوى.

لنضرب مثالًا مبالغًا فيه، فكِّر ماذا سيحدث إذا لم يكن المخطَّط البياني يحتوي على أوزان سالبة فحسب، بل يحتوي على دورة، حيث مجموع الحواف يكون عددًا سالبًا؛ أي دورة سالبة. عندئذٍ «ما من خوارزمية» ستجد أقصر المسارات في المخطط البياني «نظرًا لعدم وجود مسارات». إذا كانت لدينا دورة سالبة، يمكننا الدوران مرارًا وتكرارًا حول حواف المخطط، وفي كل مرة سيتقلص طول المسار. يمكننا الاستمرار على

هذا المنوال إلى الأبد، وسينتهي المسار عبر الدورة إلى عددٍ لا نهائي سالب. يطلِق علماء الكمبيوتر والمبرمجون اسمًا على ما يحدث عندما نضع شيئًا في برنامجٍ لا يعطي معنًى له، وهو: «المدخلات الخاطئة تعطي مخرجات خاطئة». والأمر يعتمد على الإنسان في الكشف عن المدخلات الخاطئة ومعرفةِ ما ينبغي استخدامه ومتى. يوجد جزء مهم في الدورات التدريبية في الجامعات عن الخوارزميات مخصَّص تحديدًا لتعليم علماء الكمبيوتر الناشئين ما ينبغي لهم استخدامه ومتى.

الفصل الثالث

البحث

إن حقيقة أن الخوارزميات يمكن أن تنجز أنواع المهام كافة — بدايةً من ترجمة النصوص إلى قيادة السيارات — قد تعطينا صورة مضللة عن المجالات التي يغلب فيها استخدام الخوارزميات. قد تبدو الإجابة عادية. فلا يُحتمل أن تجد برنامجًا على جهاز كمبيوتر ينجز أيَّ مهمة مفيدة على الإطلاق من دون توظيف خوارزميات مختصة بالبحث في البيانات.

السبب في ذلك أن البحث بشكلٍ أو بآخر يظهر في كل السياقات تقريبًا. البرامج تستوعب البيانات، وغالبًا ستحتاج إلى البحث عن شيء في تلك البيانات؛ ومِن ثَم فلن يكون هناك مناص من استخدام خوارزمية بحث. فالبحث ليس عملية متكررة في البرامج فحسب؛ بل إنه نظرًا لكثرة حدوثها، قد يكون البحث أكثرَ العمليات استنفادًا للوقت في تطبيقٍ ما. ويمكن لخوارزمية البحث الجيدة أن تؤدي إلى تحسيناتٍ كبيرة من حيث السرعة.

> البحث بشكل أو بآخر يظهر في كل السياقات تقريبًا … ويمكن لخوارزمية البحث الجيدة أن تؤدي إلى تحسيناتٍ كبيرة من حيث السرعة.

يتضمَّن البحث التنقيب عن عنصر معيَّن وسط مجموعة من العناصر. وهذا الوصف العام للمسألة يتضمَّن عدة تباينات. فثمَّة فارق كبير بين إذا ما كانت العناصر مرتَّبة بطريقةٍ مرتبطة بعملية البحث أم كان الترتيب عشوائيًّا. ثمة سيناريو مختلف يحدث عندما تُعطى لنا العناصر واحدًا تلو الآخر ويكون علينا أن نقرِّر ما إن كنا وجدنا العنصر الصحيح فور رؤيته أم لا من دون أن نستطيع إعادة التفكير في قرارنا. إذا بحثنا مرارًا في مجموعةٍ من العناصر، فمن المهم أن نعرف هل توجد عناصر أكثر شيوعًا من غيرها أم لا بحيث ينتهي بنا المقام إلى البحث عنها أكثر. سنتناول كل تلك التباينات في هذا الفصل،

ولكن تذكَّر أن هناك المزيد. على سبيل المثال، لن نعرض سوى مسائل «البحث الدقيق» ولكن يوجد العديد من التطبيقات التي نحتاج فيها إلى «بحث تقريبي». لنضرب مثالًا بالتدقيق الإملائي: عندما تخطئ في كتابة كلمةٍ ما، سيضطر المدقِّق اللغوي إلى البحث عن كلمات مشابهة للكلمات التي أخفق في التعرُّف عليها.

كلما زادت أحجام البيانات، زادت أهمية القدرة على البحث بكفاءة في عدد ضخم من العناصر أكثر وأكثر. وسنرى أنه إذا كانت العناصر مرتَّبة، يمكن حينها أن يسير البحث جيدًا إلى أقصى درجة. في الفصل الأول، ذكرنا أنه يمكن العثور على عنصرٍ من بين مليار عنصر مرتَّب في غضون نحو ٣٠ عملية بحث؛ وسنرى الآن كيف يمكن القيام بذلك فعليًّا.

وأخيرًا، سوف تقدِّم لنا خوارزمية البحث لمحةً عن المخاطر الخفية عندما ننتقل من مرحلة الخوارزميات إلى مرحلة التنفيذ الفعلي على برنامج الكمبيوتر الذي يعمل في نطاق حدود آلة معينة.

البحث عن إبرة في كومة قش

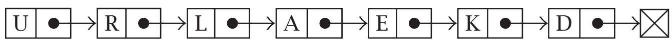
أبسط طرق البحث هي ما نقوم به للعثور على إبرةٍ في كومة قش كما يقول المثل الدَّارج. إذا أردنا العثور على عنصرٍ في مجموعة عناصر ولا توجد بنية محدَّدة تمامًا في تلك المجموعة، فلا سبيل أمامنا سوى التحقُّق من كل عنصرٍ تلو الآخر إلى أن نجد العنصر الذي نبحث عنه أو نخفق في إيجاده بعد فرز كل العناصر واستنفادها.

إذا كان لديك مجموعة من أوراق اللعب وتبحث عن ورقةٍ بعينها وسطها، يمكنك البدء بسحب الأوراق من الأعلى إلى أن تجد البطاقة التي تبحث عنها أو حتى نفاد الأوراق. وبدلًا من ذلك، يمكنك البدء في سحب البطاقات الواحدة تلو الأخرى من أسفل رِزمة أوراق اللعب. بل يمكنك أيضًا سحب الأوراق من مواضعَ عشوائيةٍ في المجموعة. فالمبدأ واحد.

لا نتعامل عادةً مع عناصر مادية في أجهزة الكمبيوتر، بل مع تمثيلات رقمية لها. وتأتي إحدى الطرق الشائعة لتمثيل مجموعات من البيانات على جهاز كمبيوتر في شكل «قائمة». والقائمة عبارة عن بنية بيانات تحتوي على مجموعةِ عناصر بطريقةٍ تمكِّننا من العثور على عنصر بناءً على العنصر الذي يسبقه. عادة ما يمكننا اعتبار أن القائمة تحوي «عناصر مترابطة»، حيث يشير كل عنصر إلى العنصر التالي حتى النهاية بحيث لا يشير العنصر الأخير إلى شيء. والمجاز ليس بعيدًا عن الواقع؛ لأن أجهزة الكمبيوتر تستخدم مواقع الذاكرة الداخلية لتخزين العناصر. وفي «القائمة المترابطة»، يحتوي كل عنصر على

مكوِّنين هما: حمولة البيانات وموقع الذاكرة للعنصر التالي في القائمة. يُطلق على الموقع في الذاكرة الذي يحمل موقع الذاكرة لمكانٍ آخرَ في الذاكرة «المؤشر». لذلك يحتوي كل عنصر في القائمة المترابطة على مؤشرٍ للعنصر التالي. يُطلق على العنصر الأول في القائمة «رأس» القائمة. ويُطلق على العناصر في القائمة اسم «العُقَد» أيضًا. ولا تشير العقدة الأخيرة إلى أي عنصر آخر؛ ومِن ثَم نقول إنها تشير إلى «قيمة فارغة»؛ أي لا شيء على الكمبيوتر.

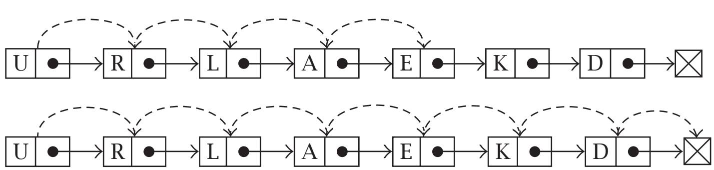
القائمة عبارة عن سلسلة عناصر، ولكن لا يلزم ترتيب السلسلة باستخدام معيار معيَّن. على سبيل المثال، فيما يلي قائمة تحتوي على بعض الحروف الأبجدية:



إذا لم تكن القائمة مرتَّبة، فستسير خطوات الخوارزمية للبحث عن عنصرٍ ما بها على النحو التالي:

(١) الانتقال إلى رأس القائمة.
(٢) إذا كان هذا هو العنصر الذي نبحث عنه، فعليك الإبلاغ بأنه قد تم العثور عليه وأوقف البحث.
(٣) الانتقال إلى العنصر التالي في القائمة.
(٤) إذا وصلنا إلى قيمة فارغة، يتم الإبلاغ بأنه لم يتم العثور على العنصر ويتوقف البحث. وإذا لم نصل إلى قيمة فارغة، نعود إلى الخطوة رقم ٢.

يُطلق على تلك العملية «البحث الخطي» أو «البحث التسلسلي». لا يوجد شيء خاص في ذلك؛ فهو تطبيق مباشر لفكرة فحص كل عنصر تباعًا إلى أن نعثر على العنصر الذي نريده. في الواقع أن الخوارزمية تجعل الكمبيوتر يقفز من مؤشرٍ إلى آخر حتى يصل إلى العنصر الذي نبحث عنه أو يصل إلى قيمة فارغة. وفيما يلي، نوضِّح ما يحدث عندما نبحث عن الحرف E أو X:



إذا كنَّا نبحث في عدد n من العناصر، فأفضل شيء يمكن أن يحدث هو الوصول إلى العنصر الذي نبحث عنه في الحال، وهو ما سيحدث إذا كان العنصر هو رأس القائمة. أما أسوأ شيء يمكن أن يحدث، فهو أن يكون العنصر في نهاية القائمة أو ليس على القائمة على الإطلاق. عندئذٍ، سنضطر إلى تصفُّح كل عناصر n. وهكذا يصبح أداء البحث التسلسلي بالقيمة $O\,(n)$.

لا يوجد ما يمكن فعله لتحسين الموقف في ذلك الوقت إذا كانت العناصر مدرجة في قائمة بتسلسل عشوائي. بالرجوع إلى أوراق اللعب، يمكن أن ترى السبب في حدوث ذلك: فلا سبيل لمعرفة مكان البطاقة التي نريدها مسبقًا إذا كانت الأوراق مختلطة جيدًا.

يواجه الناس مشكلةً في ذلك في بعض الأحيان. إذا كنا نبحث عن ورقةٍ بين كومة كبيرة من الأوراق، فربما نشعر بالضجر من تصفُّح الأوراق الواحدة تلو الأخرى. بل ربما نفكِّر في مدى تعاسة الحظ التي سنُمنَى بها إن تبيَّن أن الورقة في نهاية كومة الورق! لذا نتوقَّف عن تصفُّح كومة الورق بالترتيب ونلقي نظرة خاطفة على نهاية الكومة. لا خطأ في إلقاء نظرة خاطفة على نهاية الكومة، ولكن الخطأ أن نعتقد أن تلك الطريقة تحسِّن من فرص الانتهاء من البحث بسرعة. إذا كانت الكومة ذات ترتيب عشوائي، فلا يوجد سببٌ لكي نعتقد أن الورقة التي نبحث عنها ليست في البداية أو النهاية أو جهة اليمين أو في المنتصف. كل المواضع محتملة على نحوٍ متساوٍ؛ لذا فالبدء من قمة الكومة وصولًا إلى قاعدتها يُعَد استراتيجية جيدة مثل أي استراتيجية أخرى تضمن تفقُّد كل عنصر مرة واحدة فقط. لكن عادة ما يكون الأسهل أن نتتبَّع ما بحثنا فيه إذا كنا نبحث بترتيب معيَّن بدلًا من التنقُّل بين العناصر بلا نظام؛ ولذا نفضِّل الالتزام بطريقة البحث التسلسلي.

كل هذا يصح ما دام لا يوجد سببٌ للتشكيك في أن العنصر الذي نبحث عنه في موضعٍ معيَّن. ولكن إذا لم يكن ذلك صحيحًا، فالأمور تتغيَّر ويمكننا الاستفادة من أي معلومات إضافية قد تكون متوافرة لدينا لتسريع عملية البحث.

تأثير ماثيو والبحث

ربما لاحظت أن بعض الأشياء في مكتبٍ غير مرتَّب تجد طريقها إلى أعلى الكومة، بينما يبدو أن عناصرَ أخرى قد انزلقت إلى قاعدتها. وعند إزالة الفوضى في النهاية، يبتهج صاحب المكتب كثيرًا لأنه اكتشف أشياءَ مدفونةً وسط كومةٍ ظنَّ أنه قد فقدها منذ وقت

طويل. ربما مرَّ آخرون أيضًا بتلك التجربة. فنحن نميل إلى وضع الأشياء التي نستخدمها كثيرًا في مكانٍ قريب منَّا؛ بينما تنزلق الأشياء القليلة الاستخدام بعيدًا عن متناول أيدينا أكثر وأكثر.

لنفترض أن لدينا كومةً من المستندات نحتاج إلى العمل عليها. المستندات غير مرتَّبة بأي حال من الأحوال. نتصفح الكومة بحثًا عن المستند الذي نحتاج إليه ونبدأ العمل عليه، ثم لا نضعه في المكان الذي وجدناه فيه، بل نضعه فوق الكومة. ثم نعود مرة أخرى إلى عملنا.

قد يحدث ألَّا نعمل بالمعدل نفسه على كل المستندات. فقد نعود إلى بعضها مرارًا، وربما لا نلتفت إلى مستندات أخرى إلا نادرًا. وإذا استمررنا في وضع كل مستندٍ أعلى الكومة بعد العمل عليه، فسنكتشف بعد فترة أن المستندات الأكثر استخدامًا ستكون قريبة من القمة، بينما المستندات الأخرى الأقل استخدامًا اتجهت إلى القاعدة. تلك الطريقة مناسبة لنا؛ لأننا نقضي وقتًا أقلَّ في تحديد أماكن المستندات الكثيرة الاستخدام، ومِن ثَم يقل وقت العمل بوجه عام.

تشير هذه الطريقة إلى استراتيجية بحث عامة يتكرَّر فيها بحثنا عن العناصر نفسها وتكون بعض العناصر أكثرَ شيوعًا في استخدامها من غيرها. وبعد العثور على العنصر، نجعله في المقدمة حتى نتمكَّن من العثور عليه أسرعَ حين نبحث عنه في المرة القادمة.

ما مدى قابلية تلك الاستراتيجية للتطبيق؟ يعتمد هذا على عدد المرات التي نلاحظ فيها مثل هذه الفروق في كثرة الاستخدام. وقد تبيَّن أن هذا يحدث كثيرًا. نعرف مقولة «الغني يزداد غنًى، والفقير يزداد فقرًا». إنها لا ترتبط بالأغنياء والفقراء فحسب. فالأمر نفسه يحدث في مجموعةٍ مذهلة من الجوانب في مختلف مجالات النشاط. وقد اتخذت الظاهرة اسمًا وهو «تأثير ماثيو» أو تأثير متَّى نسبة للآية التالية في إنجيل متَّى (٢٥: ٢٩): «لأن كلَّ مَن له يُعطى فيزدادُ، ومَن ليس له فالذي عنده يُؤخذ منه.»

تتحدَّث الآية عن السلع المادية؛ لذا لنفكر في الثروة لدقيقةٍ. لنفترض أنك تمتلك استادًا كبيرًا يسع ٨٠ ألف فرد. بإمكانك قياس متوسط أطوال الأفراد في الاستاد. قد تكون النتيجة في حدود ١٧٠ سنتيمترًا (٥ أقدام و٧ بوصات). تخيَّل أنك أخرجت فردًا بصورة عشوائية من الاستاد وأدخلت أطول شخص في العالم. هل سيختلف متوسط الطول؟ حتى إذا كان أطولُ شخص في العالم ٣ أمتار (علمًا بأن هذا الطول لم يُسجَّل حتى الآن)، فلن يتغيَّر متوسط الأطوال عن القيمة السابقة؛ فالفرق مع متوسط الأطوال السابق أقلُّ من واحد على عشرة من الملِّيمتر.

تخيَّل الآن أنك بدلًا من قياس متوسط الأطوال، تقيس متوسط الثروات. يمكن أن يكون متوسط ثروات الثمانين ألف فرد السابقين مليون دولار (نحن نفترض مجموعة ثرية). تعاود الآن إدخال شخص آخر بين أغنى أثرياء العالم. قد تبلغ ثروة ذلك الشخص ١٠٠ مليار دولار. هل تلك القيمة تُحدِث فارقًا؟ نعم، تُحدِث فارقًا، بل فارقًا كبيرًا. سيزيد المتوسط من مليون دولار إلى ٢٢٤٩٩٨٧٫٥ دولار، أو أكثر من الضعف. نعلم أن الثروة ليست موزَّعة بالتساوي على مستوى العالم، ولكن ربما لا نعلم مدى عدم التساوي في توزيعها. فالتفاوت في توزيع الثروة أكبر بكثير من القياسات الطبيعية مثل الأطوال.

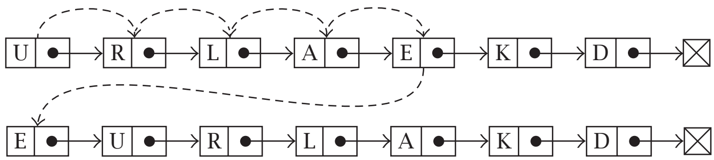
يظهر الفارق نفسه في القدرات والمَلَكات في العديد من المواقع الأخرى. فهناك العديد من الممثلين لم تسمع بهم من قبل. وهناك بضعة نجوم يظهرون في العديد من الأفلام ويربحون ملايين الدولارات. كان عالِم الاجتماع روبرت كيه ميرتون هو مَن صاغ مصطلح «تأثير ماثيو» عام ١٩٦٨ حين لاحظ أن العلماء المشهورين يحظون بالتقدير لأعمالهم أكثرَ من زملائهم الأقل شهرة حتى لو كانت إسهاماتهم متماثلة. فكلما ذاع صيت العالِم حظي بشهرةٍ أوسع.

تتَّبِع الكلمات في اللغة النمطَ نفسه؛ إذ يكون البعض منها أشهر وأكثر شيوعًا بكثير من غيره. وتضم قائمة المجالات التي تتَّسم بمثل هذه التفاوتات الصارخة حجم المدن (فالمدن الكبرى أكبر من المدن ذات الحجم العادي عدة أضعاف)، وعدد مواقع الويب وروابطها وشعبيتها (بعض المواقع لا تحظى إلا بالزوَّار العابرين بينما يحظى بعضها الآخر بملايين الزوار). كان انتشار مثل هذه التوزيعات غير المتكافئة — حيث يحصل عدد قليل من الأفراد على كم غير متناسب من الموارد — مجالًا ثريًّا للبحث والتحقيق على مدى السنوات القليلة الماضية. فالباحثون يعكفون على دراسة الأسباب والقوانين التي تقف خلف نشأة مثل تلك الظاهرة.[1]

من الممكن أن تُظهِر العناصر التي نبحث عنها مثل هذه الفروق على صعيد الشيوع والانتشار. لذا فالخوارزمية التي ستستفيد من تفاوت الشيوع بين العناصر قيد البحث تشبه إلى حدٍّ كبير وضْع كل مستند نجده أعلى الكومة:

(١) البحث عن عنصر باستخدام طريقة البحث التسلسلي.
(٢) إذا عُثر على العنصر، يُشار إلى العثور عليه ويوضع في مقدمة القائمة — على رأسها — ويتوقَّف البحث.
(٣) لو لم يُعثر على العنصر، يُشار إلى عدم وجوده، ويتوقف البحث.

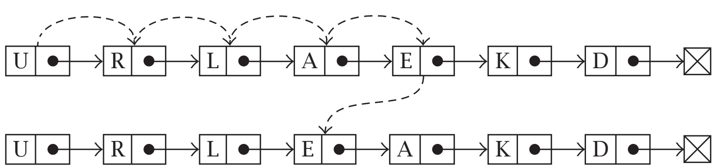
في الشكل التالي، سيؤدي العثور على الحرف E إلى جلبه إلى المقدمة:



ثمة انتقاد محتمل قد يوجَّه إلى خوارزمية «النقل إلى المقدمة» كونها ستجعل العنصر قيد البحث في المقدمة حتى لو كان البحث عنه نادرًا. وهذا صحيح، لكن إذا لم يكن العنصر كثير الاستخدام، فسينتقل تدريجيًّا إلى نهاية القائمة كلما بحثنا عن عناصر أخرى؛ لأن هذه العناصر ستنتقل بدورها إلى المقدمة. لكن يمكننا الانتباه إلى الموقف عن طريق اتباع استراتيجية أقل تطرفًا. فبدلًا من نقل كل عنصر نعثر عليه إلى المقدمة فجأة، يمكننا تحريكه موضعًا واحدًا إلى الأمام. وهذه الطريقة تسمَّى «طريقة تبديل الموضع»:

(١) ابحث عن عنصر باستخدام بحث تسلسلي.
(٢) إذا عُثر على العنصر، فأشِر إلى أنه قد عُثر عليه وأبدِله مع العنصر الذي قبله (إذا لم يكن هو العنصر الأول) وتوقَّف عن البحث.
(٣) إذا لم يُعثر على العنصر، فأشِر إلى عدم وجوده وأوقف البحث.

بهذه الطريقة، ستتحرَّك العناصر الأكثر شيوعًا تدريجيًّا إلى المقدمة، بينما ستنتقل العناصر الأقل شيوعًا إلى الخلف دون حدوث اضطرابات مفاجئة.



تُعَد طريقتا النقل إلى المقدمة وتبديل الموضع مثالين على «البحث الذاتي التنظيم»، الذي جاءت تسميته تلك من كون قائمة العناصر تُرتَّب مع تواتر عمليات البحث، وسوف تعكس مدى شيوع العناصر قيد البحث. وبناءً على مدى شيوع العنصر وسط العناصر

الأخرى، قد يكون التوفير في الوقت كبيرًا. فبينما يمكن توقُّع أداء البحث التسلسلي بالقيمة $O\left(n\right)$، يمكن أن يكون أداء البحث ذاتي التنظيم بطريقة النقل إلى المقدمة بالقيمة $O\left(n/lgn\right)$. إذا كان لدينا ما يقرب من مليون عنصر، تكون هذه القيمة هي الفرق بين مليون و٥٠٠٠٠ تقريبًا. قد تؤتي طريقة تبديل الموضع نتائجَ أفضل، لكنها تتطلب مزيدًا من الوقت لتحقيقها. والسبب في ذلك أن الطريقتين كلتيهما تتطلبان «فترة إحماء» تعلِن خلالها العناصرُ الشائعة عن نفسِها وتشقُّ طريقها نحو المقدمة. في طريقة النقل إلى المقدمة تقصُر فترة الإحماء، أما في طريقة تبديل الموضع فيستغرق الإحماء فترةً أطول، ولكننا نحصل على نتائجَ أفضل.[2]

كيبلر والسيارات والسكرتيرات

بعدما فقد عالِم الفلك الشهير يوهانس كيبلر (١٥٧١–١٦٣٠) زوجتَه بسبب الكوليرا عام ١٦١١، بدأ التخطيط للزواج مرةً أخرى. ونظرًا لكونه رجلًا منهجيًّا، لم يترك شيئًا للصدفة. وفي خطابٍ طويل أرسله إلى البارون ستراهليندورف، وصف العملية التي اتَّبعها. خطَّط كيبلر لإجراء مقابلات مع ١١ عروسًا محتملة قبل أن يتخذ قراره. وانجذب بشدة إلى المرشَّحة الخامسة، ولكنه انصرف عنها بسبب أصدقائه الذين اعترضوا على مكانتها الاجتماعية المتواضعة. ونصحوه بأن يعيد النظر في المرشَّحة الرابعة بدلًا منها. ولكنها رفضته. في النهاية، وبعد دراسة ١١ مرشَّحة، تزوَّج كيبلر المرشَّحة الخامسة، وهي سوزانا روتينجر البالغة من العمر ٢٤ عامًا.

تلك القصة القصيرة مثالٌ مطول لعملية بحث؛ فقد كان كيبلر يبحث عن اختيار مثالي بين مجموعة من المرشَّحات المحتملات. لكن كانت هناك مشكلة في عملية البحث ربما لم ينتبه لها عندما بدأ، وهي أنه ربما لا يتمكَّن من العودة إلى اختيار محتمَل بعدما رفضه.

يمكننا إعادة صياغة المسألة بمصطلحات أحدث، من خلال البحث عن أفضل طريقة لتحديد سيارة لشرائها. قرَّرنا من قبل أن نزور عددًا محدَّدًا من معارض تجارة السيارات. ولكن الاعتزاز بالنفس لن يسمح لنا بالعودة إلى معرض سيارات بعد مغادرته. إذا رفضنا سيارة، فإن حفْظ ماء الوجه أمرٌ بالغ الأهمية، ولذا لا يمكننا العودة ونقول إننا قد غيَّرنا رأينا. أو ربما يدخل شخصٌ آخر ويشتري السيارة بعدما نغادر. وأيًّا كان الأمر، علينا أن نتخذ قرارًا نهائيًّا عند كل معرض، سواء بشراء السيارة أو عدم شرائها، عدم العودة إليه.

يُعَد هذا مثالًا على «مسألة التوقُّف الأمثل». علينا أن نتخذ إجراءً وفي الوقت نفسه نحاول تعظيمَ عائدٍ ما أو تقليل تكلفةٍ ما. في هذا المثال، نريد أن نقرِّر شراء السيارة، حين يكون هذا القرار سيفضي إلى شراء أفضل سيارة يمكن شراؤها. إذا قرَّرنا في مرحلةٍ مبكرة للغاية، فربما نستقر على سيارةٍ أسوأ من السيارات التي لم نرَها بعد. وإذا قرَّرنا في مرحلةٍ متأخرة للغاية، قد نغتم حين نكتشف أننا قد رأينا أفضلَ سيارة ولكننا فقدناها. إذن متى يكون الوقت الأمثل للتوقُّف واتخاذ القرار؟

وُصِفت المشكلةُ نفسها بمزيد من الحدة تحت اسم «مسألة السكرتيرة». أنت تريد اختيار سكرتيرة من بين مجموعةٍ من المرشَّحات. يمكنك إجراء مقابلة مع المرشَّحات واحدة بواحدة. لكن يجب أن تتخذ قرارًا بالتوظيف من عدمه في نهاية كل مقابلة. وإذا رفضت مرشَّحة، لا يمكنك أن تغيِّر رأيك لاحقًا وتقدِّم لها عرضًا (قد تكون المرشَّحة جيدة للغاية ومِن ثَم ينتزعها شخص آخرُ للعمل لديه). كيف ستختار المرشَّحة؟

الإجابة بسيطة إلى حد الذهول. تجري المقابلة مع ٣٧ بالمائة من المرشَّحات وترفضهن جميعًا، ولكن مع الاحتفاظ بسجلٍ للأفضل من بينهن كمعيار استرشادي. يظهر العدد ٣٧ — الذي يبدو سحريًّا — لأن ٣٧٪ تساوي تقريبًا $1/e$، حيث e تعبِّر عن عدد أويلر، الذي يساوي تقريبًا ٢٫٧١٨٢ (تعرَّفنا على عدد أويلر في الفصل الأول). ثم تجري المقابلات مع بقية المرشَّحات. تتوقَّف عند أول مرشحة تكون أفضل من المعيار الذي معك. هذه ستكون اختيارك. وبالصيغة الخوارزمية، إذا كان لديك n من المرشَّحات:

(١) احسب n/e، لإيجاد نسبة الـ ٣٧٪ من العدد n من المرشَّحات.

(٢) ادرس أول n/e من المرشَّحات وارفضهن. ستستخدم أفضل واحدة منهن كمعيار استرشادي لك.

(٣) واصل مع بقية المرشَّحات. اختر أول مرشَّحة تجدها أفضل من المعيار الذي معك، وتوقَّف.

لن تجد الخوارزمية المرشَّح الأفضل على الدوام؛ ففي النهاية، قد يكون المرشَّح الأفضل عمومًا هو المرشَّح المعياري الذي حدَّدته ضمن أول ٣٧ بالمائة، وسبق أن رفضته. ويمكن إثبات أنها ستجد لك المرشَّح الأفضل في ٣٧ بالمائة (مرة أخرى $1/e$) من جميع الحالات؛ إضافة إلى ذلك، لا توجد طريقة أخرى من شأنها أن تنجح في العثور على المرشَّح الأفضل في مزيد من الحالات. بعبارة أخرى، الخوارزمية هي أفضل الطرق التي يمكن

تتبُّعها؛ وعلى الرغم من أنها قد لا تعطيك أفضل مرشَّح في ٦٣٪ من الحالات، فلا توجد استراتيجية أخرى تعطيك نتيجةً أفضل من تلك النتيجة.

بالعودة إلى السيارات، لنفترض أنك قرَّرت الذهاب إلى ١٠ معارض لتجارة السيارات. ينبغي أن نذهب إلى أول أربعة معارض وندوِّن ملحوظةً بأفضل عرض مقدَّم منها، دون شراء. بعد ذلك نبدأ في زيارة المعارض الستة المتبقية، وسنشتري من أول معْرض يقدِّم لنا عرضًا أفضلَ من العرض الذي دوَّناه (وسنتجاوز البقية حينئذٍ). ربما نكتشف أن المَعارض الستة جميعًا تقدِّم عروضًا أسوأ من الأربعة الأولى التي ذهبنا إليها من دون أن نشتري. ولكن لا توجد استراتيجيةٌ أفضلُ يمكن أن تعطينا احتمالاتٍ أفضل للحصول على أفضل صفقة.

لقد افترضنا أننا نريد العثور على أفضلِ مرشَّح محتمَل ولن نقبل بأقلَّ من ذلك. ولكن ماذا لو استطعنا القبول بشيءٍ أقلَّ في الحقيقة؟ هذا يعني أنه على الرغم من أننا نريد حقًّا أفضلَ سكرتيرة أو سيارة، فربما نقنع باختيارٍ آخرَ، وربما نسعد به وإن لم يكن قدْر سعادتنا لو اخترنا الأفضل. إذا وضعنا إطارًا للمسألة بهذا الشكل، فإن أفضلَ طريقة نحدِّد بها اختيارنا هي استخدام الخوارزميةِ نفسها السالفةِ الذكر، ولكن مع اختبار الجذر التربيعي لعدد المرشَّحين — $\sqrt{n}$ — وإسقاطه. إذا فعلنا ذلك، فسوف تزيد احتمالية انتقاء الخيار الأفضل مع زيادة عدد المرشَّحين؛ فكلما زادت قيمة n، ارتفعت احتمالية انتقاء الخيار الأفضل إلى ١ (أي ١٠٠ بالمائة).[3]

البحث الثنائي

تناولنا طرقَ البحث المختلفة وطابقناها مع مختلف السيناريوهات. والقاسم المشترك في كل هذه الطرق أن العناصر التي نتناولها تأتينا دون ترتيب محدَّد؛ وفي أفضل الحالات، نرتِّبها نحن تدريجيًّا حسب تكرار استخدامها في عمليةِ بحث ذاتية التنظيم. ولكن يتغيَّر الموقف تمامًا إذا كانت العناصر مرتَّبة منذ البداية.

لنفترض أن لدينا كومة من المستندات، كل مستند معرَّف برقم. المستندات في الكومة مرتَّبة حسب الرقم التعريفي من الأصغر إلى الأكبر (لا يلزم أن تكون الأرقام متوالية). إذا كان لدينا كومة كتلك ونبحث عن مستندٍ ذي رقم تعريفي محدَّد، فمن الحماقة أن نبدأ بأول مستند ونستمر إلى نهاية الكومة حتى نعثر على المستند الذي نبحث عنه. هناك استراتيجيةٌ أفضل بكثير وهي الانتقال مباشرة إلى منتصف الكومة. ثم نطابق الرقم

التعريفي الوارد على المستند في المنتصف برقم المستند الذي نبحث عنه. هناك ثلاث نتائج محتملة:

(١) إذا حالفنا الحظ، قد نقع على المستند الذي نريده بالضبط. وهنا تنتهي عملية البحث.

(٢) أن يكون الرقم التعريفي للمستند الذي نبحث عنه أكبر من الرقم التعريفي للمستند الذي بين أيدينا. وفي تلك الحالة نعلم يقينًا أنه يمكننا تجاهل المستند الذي بين أيدينا «وكذلك المستندات التي قبله». وحسب ترتيب المستندات، ستكون الأرقام التعريفية لها جميعًا أصغر. وهذا يعني أننا لم نصل إلى هدفنا بعدُ.

(٣) أن يحدث العكس: أن يكون الرقم التعريفي للمستند الذي نبحث عنه أصغر من الرقم التعريفي للمستند الذي بين أيدينا. وعندئذٍ يمكننا تجاهل المستند الذي بين أيدينا مطمئنين، «وكذلك كل المستندات التي تأتي بعده». وعندئذٍ نكون قد تجاوزنا هدفنا.

في أيٍّ من النتيجتين الأخيرتين، يصبح المتبقي لدينا كومة تعادل نصف الكومة الأصلية على أقصى تقدير. إذا بدأنا بعددٍ فردي من المستندات، ولنقل العدد n، فإن ناتج قسمة العدد n من المستندات إلى نصفين يعطينا جزأين، كل جزء يحتوي على العدد $n/2$ من العناصر (بغض النظر عن الجزء الكسري في عملية القسمة):

$$\circ \quad \circ \quad \times \quad \circ \quad \circ$$

أما إذا كان لدينا عدد زوجي من العناصر، فسيعطينا ناتج القسمة جزأين؛ أحدهما يضم $n/2 - 1$ من العناصر، والآخر يضم $n/2$ من العناصر:

$$\circ \quad \times \quad \circ \quad \circ$$

ما زلنا لم نعثر على ما نبحث عنه بعد، ولكننا في موقفٍ أفضلَ من ذي قبل بكثير؛ فقد صار لدينا عدد أقل بكثير من العناصر للبحث فيها. وهذا ما نفعله. نتحقق من المستند الأوسط في «العناصر المتبقية» ونكرِّر الإجراء.

في الشكل في الصفحة التالية، يمكنك أن ترى كيف تتطوَّر العملية بالنسبة إلى ١٦ عنصرًا نبحث فيها عن العنصر رقم ١٣٥. نحدِّد الحدود التي نبحث بداخلها والعنصر الأوسط باللون الرمادي.

في البداية، يكون نطاق البحث هو مجموعة العناصر كلها. ننتقل إلى العنصر الأوسط، الذي نجد أنه العنصر ٣٨٤. هذا العنصر أكبر من ١٣٥، ومِن ثَم نتجاهله، وكذلك كل العناصر الواقعة إلى يمينه. نأخذ العنصر الأوسط في العناصر المتبقية وهو العنصر ٧٢. هذا العنصر أصغر من ١٣٥، لذا نتجاهله وكذلك كل العناصر الواقعة على يساره. وبذلك يكون نطاق البحث قد تقلَّص إلى ثلاثة عناصر فقط. نأخذ العنصر الأوسط ونجد أنه العنصر المطلوب. لم يستغرق الأمر سوى ثلاث عمليات تحقُّق لإنهاء بحثنا، ولم نحتَج إلى التحقُّق حتى من ١٣ عنصرًا من بين ١٦ عنصرًا.

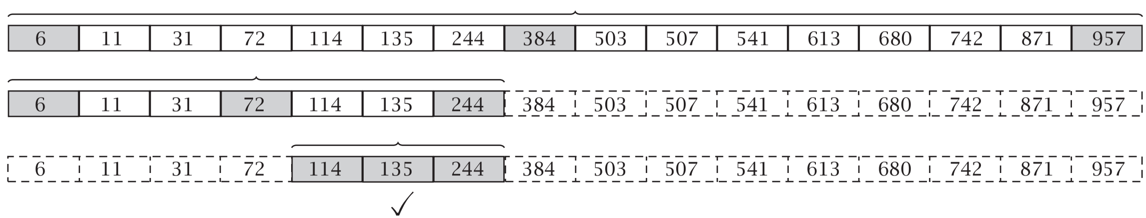
تصلح تلك الطريقة أيضًا إذا كنا نبحث عن شيءٍ غير موجود. يمكنك إدراك ذلك في الشكل التالي، حيث إننا نبحث في العناصر نفسها عن عنصر باسم ٥٢٠.

في هذه المرة، ٥٢٠ أكبر من ٣٨٤، ومِن ثَم نقيِّد البحث في النصف الأيمن من تلك العناصر. وهناك نجد أن العنصر الأوسط في النصف العلوي هو ٦١٣، وهو أكبر من ٥٢٠. عندئذٍ نقيِّد البحث في ثلاثة عناصر فقط، أوسطهم هو ٥٠٧. هذا العنصر أصغر من ٥٢٠. لذا نتجاهله ليتبقى لدينا عنصر واحد للتحقُّق منه ونكتشف أنه ليس العنصر الذي نريده. ومِن ثَم يمكننا إنهاء عملية البحث ونقول إن العنصر غير موجود. لم تستغرق العملية أكثر من ٤ عمليات تحقُّق.

يُطلَق على الطريقة التي وصفناها للتو «البحث الثنائي»؛ لأن في كل مرة نقسِّم نطاق القيم التي نبحث فيها إلى نصفين. ونطلِق على نطاق القيم الذي نُجري فيه عملية البحث «مساحة البحث». باستخدام ذلك المفهوم، يمكننا تحويل البحث الثنائي إلى خوارزمية تتكوَّن من الخطوات التالية:

(١) إذا كانت مساحة البحث فارغة، فهذا يعني أنه لا يوجد مكانٌ للبحث فيه، ومِن ثَم نقول إن البحث قد فشل ونتوقَّف. أما إذا لم تكن كذلك، نبحث عن العنصر الأوسط في مساحة البحث.

(٢) إذا كان العنصر الأوسط أقلَّ من العنصر قيد البحث، نقلِّص مساحة البحث من العنصر الأوسط بترتيب تصاعدي ونعود إلى الخطوة ١.


6 11 31 72 114 135 244 384 503 507 541 613 680 742 871 957
6 11 31 72 114 135 244 384 503 507 541 613 680 742 871 957
6 11 31 72 114 135 244 384 503 507 541 613 680 742 871 957

(٣) ولكن إذا كان العنصر الأوسط أكبر من العنصر قيد البحث، نقصُر مساحة البحث على العنصر الأوسط ونعود إلى الخطوة ١.

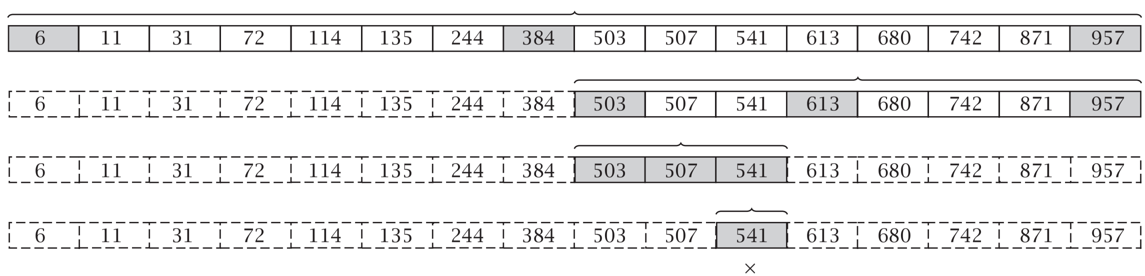
(٤) وإذا كان العنصر الأوسط مساويًا للعنصر قيد البحث، نقول إن البحث قد نجح ونتوقَّف.

بهذه الطريقة، نُقسِّم العناصر التي علينا البحث عنها على اثنين. ويُطلق على هذه الطريقة فرِّق تَسُد. تؤدي تلك الطريقة إلى تكرار عملية القسمة، ما يعطينا لوغاريتمًا كما رأينا في الفصل الأول. وتكرار القسمة على اثنين يعطينا لوغاريتمًا أساسه العدد اثنان. وفي أسوأ الحالات، سيستمر البحث الثنائي في قسمة العناصر مرارًا وتكرارًا حتى تستحيل عملية القسمة كما رأينا في مثال عملية البحث الفاشلة التي لم يُعثر فيها على العنصر. بالنسبة إلى العدد n من العناصر، لا يمكن أن تتكرَّر عملية القسمة أكثر من lgn من المرات؛ وعليه فإن قيمة تعقيد البحث الثنائي تساوي $O\ (lgn)$.

نسبة التحسُّن مقارنة ببحثٍ تسلسلي — حتى في بحث ذاتي التنظيم — نسبة رائعة. فلن يستغرق الأمر أكثر من ٢٠ استكشافًا للبحث في مليون عنصر. لننظر من زاوية أخرى، باستخدام مائة احتمال، نتمكَّن من البحث عن أي عنصر وإيجاده من بين ٢^{١٠٠} ≈ ١٫٢٧ × ١٠^{٣٠}، وهذا العدد أكبر من «نونليون».

إن فاعلية البحث الثنائي مذهلة. وربما لا يضاهي فاعليتها إلا شهرتها السيئة. فهي خوارزمية قائمة على الحدْس. ولكن هذه الطريقة البسيطة أثبتت مرارًا أنها معقَّدة وخادعة بما لا يجعلها تعمل بصورةٍ صحيحة في برنامجٍ من برامج الكمبيوتر. ولأسباب لا تتعلَّق بخوارزمية البحث الثنائي في حد ذاتها، بل بالطريقة التي نحوِّل بها الخوارزميات إلى تعليماتٍ برمجية حقيقية في لغة البرمجة، وقَع مبرمجون فريسةً لأخطاءٍ برمجية خبيثة تسللت إلى عمليات التنفيذ. والحديث ليس عن المبتدئين في المجال؛ فحتى المبرمجون العالميون فشلوا في تنفيذها بالطريقة الصحيحة.[4]

للتعرُّف على المكامن المحتملة لتلك الأخطاء، فكِّر كيف نجد العنصر الأوسط بين العناصر التي نريد البحث فيها في الخطوة الأولى من الخوارزمية. إليك فكرة بسيطة: العنصر الأوسط في العنصرين m وn هو $(m + n)\ /2$ ويُقرَّب إلى عدد صحيح إذا لم يكن الناتج عددًا طبيعيًّا. هذا صحيح ونابع من الرياضيات الابتدائية البسيطة، ولذا ينطبق في أي مجال.


6 11 31 72 114 135 244 384 503 507 541 613 680 742 871 957
6 11 31 72 114 135 244 384 503 507 541 613 680 742 871 957
6 11 31 72 114 135 244 384 503 507 541 613 680 742 871 957
6 11 31 72 114 135 244 384 503 507 541 613 680 742 871 957
×

باستثناء أجهزة الكمبيوتر. تمتلك أجهزة الكمبيوتر موارد محدودةً من بينها الذاكرة. ولذلك لا يمكن تمثيل كل الأعداد التي نريدها على جهاز الكمبيوتر. ببساطة، ستكون بعض الأرقام كبيرةً للغاية. فإذا كان الكمبيوتر له حدٌّ أعلى لحجم الأعداد التي يمكن أن يتعامل معها، فإن العنصرين m وn ينبغي أن يكونا أقلَّ من هذا الحد. بالطبع $(m + n)\ /2$ أقل من ذلك الحد. ولكن كي نحسب $(m + n)\ /2$، ينبغي أن نحسب $m + n$ أولًا ثم نقسم الناتج على اثنين، «وذلك المجموع قد يكون أكبر من الحد الأعلى»! وهذا يسمَّى «الفيض»؛ أي تجاوز نطاق القيم المسموح به. ولذلك تقع في خطأ برمجي لم تظن أنك قد تقع فيه. ولن تكون النتيجة هي القيمةَ الوسطى، بل شيئًا مختلفًا تمامًا.

لا تيأس إذا وجدت نفسك بائسًا تتأمَّل سطرًا من الشيفرة البرمجية لا يفعل ما تعتقد أنه يجب أن يفعله. لست وحيدًا في ذلك. فهو يحدث للجميع؛ بل يحدث للصفوة.

ما إن تعرف ذلك، يصبح الحل بسيطًا. أنت لم تحسب القيمة الوسطى بالصورة $(m + n)\ /2$، بل بالصورة $m + (n - m)\ /2$. النتيجة واحدة ولكن لم يحدث فيض. عندما ننظر إلى الوراء، يبدو الأمر بسيطًا. ولكن بعد أن تتضح الأمور، يهبط الإلهام على الجميع.

نحن هنا معنيون بالخوارزميات وليس بالبرمجة، ولكن اسمحوا للمؤلِّف أن يشارك بنصيحةٍ لمن يكتبون أو يريدون كتابة برامج الكمبيوتر. لا تيأس إذا وجدت نفسك بائسًا تتأمَّل سطرًا من الشيفرة البرمجية لا يفعل ما تعتقد أنه يجب أن يفعله. لا تفزع إذا أدركتَ في اليوم التالي أن الخطأ كان بالفعل أمام عينيك طوال الوقت. كيف لم ترَه؟ لست وحيدًا في ذلك. فهذا يحدث للجميع؛ بل يحدث للصفوة.

يتطلَّب البحث الثنائي فرز العناصر. لذا فحتى تجني ثماره، ينبغي أن تتمكَّن من فرز العناصر بكفاءة؛ ومن هنا ننتقل إلى الفصل التالي، حيث سنتناول كيف يمكن فرز العناصر باستخدام الخوارزميات.

الفصل الرابع

الترتيب

ينص دستور الولايات المتحدة على ضرورة إجراء إحصاء للسكان كل عشر سنوات من أجل توزيع الضرائب والنواب بين مختلف الولايات. وقد أُجري الإحصاء السكاني الأول بعد الثورة الأمريكية عام ١٧٩٠، ومنذ ذلك الحين يُجرى الإحصاء كل عشر سنوات.

في غضون مائة عام منذ عام ١٧٩٠، زاد عدد سكان الولايات المتحدة بسرعة؛ إذ قفز من أقل من ٤ ملايين نسمة بقليل في الإحصاء الأول إلى أكثر من ٥٠ مليون نسمة عام ١٨٨٠. وهنا تكمُن مشكلة؛ فقد استغرق الأمر ثماني سنوات لتعداد هؤلاء السكان. عندما حلَّت سنة الإحصاء التالية، في عام ١٨٩٠، صار عدد السكان أكبر. ولو جرى التعداد بالطريقة نفسها، لربما لم يكن لينتهي حتى حلول سنة التعداد «التالية»؛ أي ١٩٠٠.

في ذلك الوقت، كان هيرمان هوليريث — وكان حينها خريجًا شابًّا من كلية المناجم بجامعة كولومبيا (تخرَّج عام ١٨٧٩ عندما كان عمره ١٩ عامًا) — يعمل لدى مكتب الإحصاء الأمريكي. وإدراكًا منه لمشكلة ضيق الوقت، حاول إيجاد طريقة لتسريع عملية الإحصاء باستخدام الآلات. استوحى هوليريث الطريقة التي يستخدم بها قاطعو التذاكر الثقوبَ الموجودة في تذاكر القطارات لتسجيل بيانات المسافر؛ ومِن ثَم اخترع طريقة يمكن بها استخدام «بطاقات مثقبة» لتسجيل تفاصيل الإحصاء. بعد ذلك يمكن معالجة تلك البطاقات باستخدام «آلات جدولة»، وهي أجهزة كهروميكانيكية يمكن أن تقرأ البطاقات المثقبة وتستخدم البيانات المختزنة فيها من أجل إنشاء جدول إحصائي.

استُخدمت آلة الجدولة التي اخترعها هوليريث في إحصاء ١٨٩٠ وقلَّلت الوقت اللازم لإكمال الإحصاء إلى ست سنوات، وحينها تبيَّن أن تعداد سكان الولايات المتحدة بلغ نحو ٦٣ مليون نسمة. قدَّم هوليريث آلات الجدولة التي اخترعها إلى الجمعية الإحصائية الملكية، وأشار إلى «ضرورة عدم اعتبار أن ذلك النظام لا يزال في مرحلة التجارب. فقد

أُحصي ما يزيد على ١٠٠٠٠٠٠٠٠ بطاقة عدة مرات على هذه الآلات، ما أتاح فرصةً كبيرة لاختبار قدراتها.»[1] وعقب هذا الإحصاء، بدأ هوليريث شركة خاصة، تحت اسم «هوليريث لأنظمة الجدولة الكهربية». وبعد سلسلة من إعادة التسمية وعمليات الدمج، صار اسمها «المؤسسة الدولية للحاسبات الآلية» (IBM)، وذلك في عام ١٩٢٤.

في الوقت الحاضر، أصبح الترتيب سائدًا في كل مناحي الحياة حتى إنه صار غير ملحوظ إلى حدٍّ كبير. فقبل بضعة عقود، كانت المكاتب تعجُّ بخزانات الملفات التي تحتوي على مجلدات موسومة بأسماء وفريق من الموظفين معنيين بالحفاظ على تلك المجلدات بالترتيب المطلوب، مثل الترتيب الأبجدي أو الزمني. في المقابل، يمكننا الآن ترتيب الرسائل في صناديقنا البريدية بضغطة زر، ويمكننا ترتيبها باستخدام معايير مختلفة، كالترتيب حسب الموضوع أو التاريخ أو اسم المرسِل. تُرتَّب جهات الاتصال في أجهزتنا الرقمية من دون أن ننتبه لذلك؛ ونكرِّر أننا قبل بضع سنين كنا نواجه مشقَّة في التأكد من تنظيم جهات الاتصال في دفاتر يومياتنا.

نعود إلى إحصاء الولايات المتحدة، كان الترتيب واحدًا من أوائل الأمثلة على أتمتة المكتب؛ لذا ليس من المستغرب أن كان هذا المثال من أوائل تطبيقات أجهزة الكمبيوتر الرقمية. وقد طُور العديد من خوارزميات الترتيب المختلفة. بعض هذه الخوارزميات خارج نطاق الاستخدام العملي، ولكن لا يزال يوجد عدد من خوارزميات الترتيب المختلفة يشيع استخدامها بين المبرمجين؛ لأنها توفِّر عددًا نسبيًّا من المزايا والعيوب. ويُعَد الترتيب جزءًا جوهريًّا من مهام أجهزة الكمبيوتر لدرجة أن ما من كتابٍ يتناول الخوارزميات لا يخصص جزءًا منه لها، ولكن نظرًا لوجود العديد من خوارزميات الترتيب المختلفة، فإن استكشافها يتيح لنا تقديرَ جانبٍ مهم من عمل علماء الكمبيوتر والمبرمجين. على غرار صنَّاع الأدوات، يمتلك المبرمجون وعلماء الكمبيوتر صندوق أدوات كامل تحت تصرُّفهم. وقد يكون هناك أدوات مختلفة للمهمة نفسها. لنضرب مثالًا بأنواع مفكات البراغي المختلفة. لدينا على سبيل المثال لا الحصر المفكات العادية وفيليبس وألين وروبرتسون. وعلى الرغم من أن الهدف من استخدامها جميعًا واحد، فإن بعض البراغي تتطلب مفكاتٍ معينة. يمكننا استخدام المفك العادي على برغيٍّ مُصلَب في بعض الأحيان؛ ولكن يجب استخدام الأداة المناسبة للمهمة بوجه عام. الشيء نفسه ينسحب على الترتيب. فعلى الرغم من أن جميع خوارزميات الترتيب تنظِّم العناصر، فإن كل واحدة منها تناسب استخداماتٍ معينةً أكثرَ من غيرها.

قبل أن نبدأ في تناول هذه الخوارزميات، لنلقِ نظرة على بعض الإيضاحات لما تفعله هذه الخوارزميات بالتحديد. لا شك أنها ترتِّب العناصر، ولكن هذا يطرح السؤال، ما الذي نعنيه حقًّا بعبارة «ترتيب البيانات»؟

نفترض أن لدينا مجموعة من البيانات المترابطة — التي يُطلق عليها عادةً «السجلات» — تحتوي على بعض المعلومات التي تهمنا. على سبيل المثال، قد تكون تلك البيانات رسائلَ البريد الإلكتروني في صندوق الوارد لدينا. نريد إعادة تنظيم تلك البيانات بحيث تظهر بترتيبٍ معيَّن مفيد لنا. ينبغي أن تجري إعادة الترتيب باستخدام سِمة أو سمات محدَّدة في البيانات. في مثال البريد الإلكتروني، قد نرغب في ترتيب الرسائل حسب تاريخ التسلُّم، أو بترتيبٍ زمني، أو اسم المرسِل، أو بترتيب أبجدي. قد يكون الترتيب تصاعديًّا — من الرسائل الأقدم إلى الأحدث — أو تنازليًّا؛ أي من الرسائل الأحدث إلى الرسائل الأقدم. يجب أن تكون بيانات مخرجات عملية الترتيب هي نفسها بيانات المدخلات؛ بتعبير تقني، يجب أن تكون المخرجات عبارة عن «تعديل في ترتيب» البيانات الأصلية، بمعنى أن يتغيَّر ترتيب البيانات الأصلية من دون أن يتغيَّر فيها أي شيء آخر.

على الرغم من أن الهدف من استخدامها جميعًا واحد، فإن بعض البراغي تتطلب مفكاتٍ معينة ... الشيء نفسه ينسحب على الترتيب. فعلى الرغم من أن جميع خوارزميات الفرز والترتيب تنظم العناصر، فإن كل واحدة منها تناسب استخدامات معينة أكثر من غيرها.

عادة ما يُطلق على السمة التي نستخدمها في ترتيب البيانات «المفتاح». قد يكون المفتاح «مفردًا» عندما نرى أننا لا نستطيع تفكيكه إلى أجزاء، أو قد يكون «مركَّبًا» عندما يتكوَّن من أكثر من سمة واحدة. إذا أردنا ترتيب رسائل البريد الإلكتروني حسب تاريخ التسلُّم، فهذا مفتاح مفرد (لا يهم إن كان يمكن تقسيم تاريخٍ ما إلى سنوات وشهور وأيام، وربما أيضًا يحتوي على وقت التسلُّم بالتحديد). لكن ربما نرغب في فرز رسائل البريد الإلكتروني حسب اسم المرسِل، وعندئذٍ تُرتَّب كل الرسائل الواردة من هذا المرسِل حسب تاريخ التسلُّم. والجمع بين التاريخ واسم المرسِل يشكِّل مفتاحًا مركَّبًا لعملية الترتيب التي نجريها.

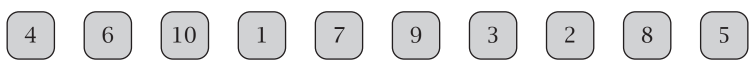
يمكن استخدام أيِّ نوع من السمات كمفتاح للترتيب ما دام يمكن ترتيب قيمها. وبالطبع ينطبق ذلك على الأرقام. إذا أردنا ترتيب بيانات المبيعات حسب رقم المبيعات

للعناصر المبيعة، يكون عدد المبيعات عبارة عن عدد صحيح. وعندما تكون المفاتيح نصية، مثل رسائل البريد الإلكتروني الواردة من المرسل، فعادةً ما يكون الترتيب الذي نريده أبجديًّا. فخوارزميات الفرز والترتيب ينبغي أن تعرف كيف تقارن بياناتنا بحيث تستنتج ترتيبها، ولكن أيُّ طريقة مقارنة فعَّالة سوف تفي بالغرض.

سنبدأ استكشافنا لطرق الترتيب باستخدام خوارزميتين قد تكونان معروفتين؛ ربما لأنهما من الخوارزميات الأكثر بديهية، بل تُستخدم من قِبل أشخاصٍ ليس لديهم معرفة بالخوارزميات عندما يكون عليهم ترتيب مجموعة من الأشياء.

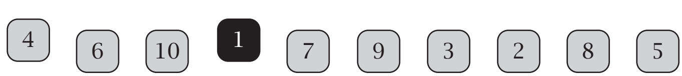
طرق الترتيب البسيطة

مهمتنا هي ترتيب العناصر التالية:

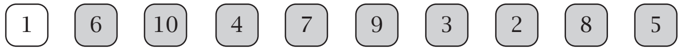


لا شك أنك إذا ألقيت نظرةً على المهمة، فستجدها تافهةً للغاية؛ إنها الأعداد من واحد إلى عشرة. ولكن تبسيط الأمور سيتيح لنا التركيز على المنطق الذي تُقام عليه مهمة الترتيب.

أولًا: نطَّلع على جميع العناصر ونبحث عن العدد الأصغر. نأخذه من مكانه حيث وجدناه ونضعه في المقدمة. العدد الأصغر هنا هو العدد ١، ولذا ينبغي أن يوضع في الموضع الأول. وبما أن هذا الموضع يشغله عنصر آخر، لا بد أن نفعل شيئًا مع العدد ٤ الذي يقع في الموضع الأول حاليًّا، ولا يمكن أن نكتفي بحذفه. ما يمكننا فعله هو تبديله مع العدد الأصغر؛ أي ننقل العدد الأصغر إلى الموضع الأول وننقل العدد الذي كان في الموضع الأول سابقًا إلى الموضع الذي تُرك شاغرًا عند نقل العدد الأصغر. وهكذا ننتقل من هذا الموضع حيث العدد الأصغر مظلَّل باللون الأسود،

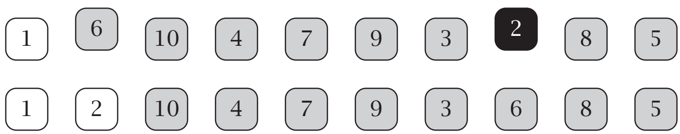


وننقله إلى هذا الموضع،

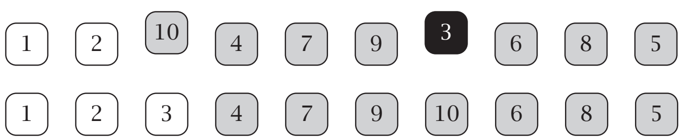


حيث العدد الأصغر مظلَّل باللون الأبيض كي نشير إلى أنه في موضعه الصحيح حسب الترتيب.

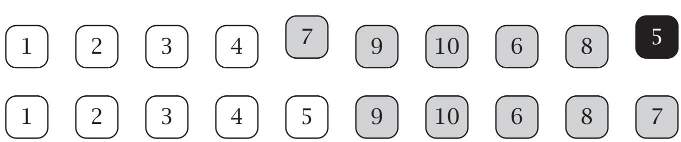
نفعل الأمر نفسه مع جميع الأعداد باستثناء العدد الأصغر الذي وجدناه؛ أي مع جميع الأعداد من الموضع الثاني بترتيب تصاعدي (الأعداد المظللة باللون الرمادي). نبحث عن العدد الأصغر بينها، وهو ٢، ونبدله مرة أخرى مع أول عدد من الأعداد التي «لم تُرتَّب» وهو العدد ٦:



نكرِّر الأمر نفسه مرة أخرى. نتعامل مع العناصر بدءًا من العنصر الثالث بترتيب تصاعدي؛ نبحث عن العدد الأصغر، وهو العدد ٣، ونبدله مع العنصر الموجود في الموضع الثالث وهو العدد ١٠:



إذا استمررنا بهذه الطريقة، فسيظل العدد ٤ في مكانه لأنه في الموضع الصحيح بالفعل، وسننتقل إلى العدد ٥ كي نضعه في موضع ترتيبه الصحيح:



في كل نقطة يقل عدد العناصر التي نمرُّ عليها لإيجاد العدد الأصغر أكثر وأكثر. في النهاية، سنوجِد العدد الأصغر لآخِر عنصرين، وبمجرد الانتهاء من ذلك، يكتمل ترتيب جميع العناصر.

يُطلق على طريقة الترتيب تلك اسم «الترتيب الانتقائي»؛ لأننا في كل مرة نختار العنصر الأصغر من بين العناصر التي لم تُرتَّب ونضعه حيث ينبغي أن يكون. ومثل جميع خوارزميات الفرز والترتيب التي سنتناولها، لا تواجه خوارزمية الترتيب الانتقائي مشكلةً مع الروابط؛ أي العناصر التي لها الترتيب نفسه. فإذا وجدنا أكثر من عنصر أصغر عند فحص العناصر غير المرتَّبة، نختار أيًّا منها باعتباره العنصر الأصغر. وفي المرة القادمة سنبحث عن العنصر المرتبط ونضعه بجوار العنصر المساوي له.

خوارزمية الترتيب الانتقائي من الخوارزميات البسيطة والمباشرة. فهل هي خوارزمية جيدة أيضًا؟ إذا انتبهنا إلى ما نفعله، فسنجد أننا ننتقل من بداية العناصر التي نريد ترتيبها إلى نهايتها، وفي كل مرة نحاول البحث عن العنصر الأصغر من بين العناصر غير المُرتَّبة. فإذا كان لدينا العدد n من العناصر، فإن تعقيد خوارزمية الترتيب الانتقائي يساوي $O\left(n^2\right)$. وهذا ليس سيئًا في ذاته؛ فدرجةُ التعقيد تلك ليست مانعة، ويمكننا حل مسائل كبيرة (مثل ترتيب عدد كبير من العناصر) في مدة زمنية معقولة.

تتمثَّل المسألة تحديدًا في وجود خوارزمياتٍ أسرعَ من تلك؛ لأن الترتيب عملية بالغة الأهمية. لذا فعلى الرغم من أن خوارزمية الترتيب الانتقائي ليست سيئة بطبيعتها، فإننا عادة ما نفضِّل استخدام خوارزميات أخرى أكثر تطورًا، عندما يكون لدينا عدد كبير من العناصر. في الوقت نفسه، فإن خوارزمية الترتيب الانتقائي ليست سهلة الفهم على البشر فحسب، بل يسهل تنفيذها على جهاز الكمبيوتر بطريقة فعَّالة. لذا من الواضح أنها ليست ذات أهمية أكاديمية فحسب، بل إنها تُستخدم في الجانب العملي كثيرًا.

يمكن أن ينسحب الأمر نفسه على خوارزمية ترتيب بسيطة أخرى سنتناولها فيما يلي. كما هو الحال مع الترتيب الانتقائي، فإن هذه الطريقة في الترتيب سهلة الفهم بعيدًا عن أجهزة الكمبيوتر. في الحقيقة، إنها الطريقة التي قد نرتِّب بها البطاقات في أيدينا في ألعاب الورق.

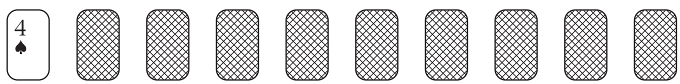
تخيَّل أنك تمارس إحدى ألعاب الورق وأخذت فيها عشر أوراق (كأن تلعب لعبة الرامي على سبيل المثال). عندما تأخذ ورقةً تلو الأخرى، تريد ترتيبها في يدك. لنفترض أن ترتيب الأوراق، من الأصغر إلى الأكبر، هو:

الترتيب

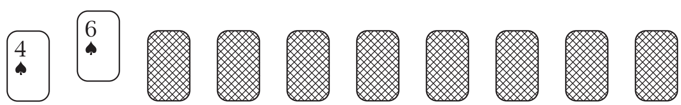
2 3 4 5 6 7 8 9 *J* *Q* *K* *A*

في الحقيقة، في العديد من الألعاب (وفي لعبة الرامي) يمكن أن تكون ورقة الآس هي الورقة ذات الترتيب الأقل والأعلى، لكن سنفترض أنه لا يوجد سوى ترتيب واحد.

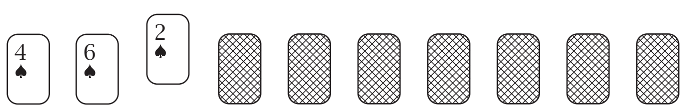
ستلعب بكل ورقة، بحيث تبدأ بورقة واحدة في يدك ويبقى تسع أوراق تأتي تباعًا كما يلي:



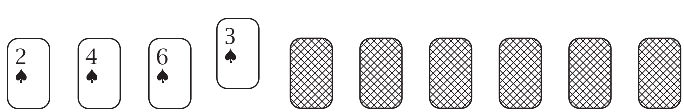
الآن، تأتي إلى الورقة الثانية، وهي الرقم ستة:



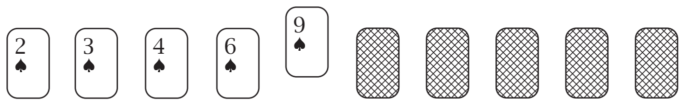
مكان ورقة الستة جيد بجوار بطاقة الأربعة، ولذا تتركها وتأخذ ورقة أخرى ويتبيَّن أنها اثنان:



هذه المرة، وحتى تُبقِي الأوراق مرتَّبة في يدك، تحتاج إلى نقل الورقة اثنين إلى يسار الورقة أربعة، ومِن ثَم تتحرك الورقة أربعة والورقة ستة موضعًا واحدًا جهة اليمين. ويكون ذلك قبل أن تلعب بورقة أخرى، وهي ثلاثة:



تُدخل الورقة ثلاثة بين الورقتين اثنين وأربعة، وتبحث عن الورقة التالية، وهي الورقة تسعة. تلك الورقة في المكان الصحيح بالفعل في يدك.



يمكن أن تستمر في ترتيب الأوراق في يدك، مثل الأوراق ٧ وQ وJ و٨ و٥. في النهاية، سينتهي بك الأمر إلى أوراقٍ مرتَّبة في يدك.

لقد أُدخِلت كل ورقة جديدة في الموضع الصحيح بالنسبة إلى الأوراق السابقة التي وُزِّعت. لذلك تسمَّى هذه الطريقة «الترتيب بالإدراج» وتصلح لأي نوع من العناصر، وليس أوراق اللعب فقط.

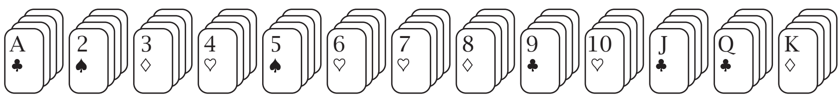
ومثل خوارزمية الترتيب الانتقائي، تتسم خوارزمية الترتيب بالإدراج بالبساطة في تنفيذها. وتبيَّن أن لها درجةَ التعقيد نفسها: $O\left(n^2\right)$. ولكن لها خاصية تميزها؛ فكما رأينا في مثال لعبة الأوراق، «لست بحاجة إلى معرفة العناصر سلفًا قبل ترتيبها». في الواقع إن ترتيب العناصر يتم وقت الحصول عليها. وهذا يعني أنه يمكنك استخدام خوارزمية الترتيب بالإدراج عندما تتدفَّق إليك العناصر المراد ترتيبها مباشرة. لقد قابلنا هذا النوع من الخوارزميات التي تطبَّق مباشرة مع توفير المدخلات عندما تناولنا مسألة جدولة المسابقات في التمثيلات البيانية بالفصل الثاني، وأسميناها «الخوارزمية الفورية». فإذا كان علينا ترتيب عدد غير معروف من العناصر، أو إذا كان لا بد أن نستطيع التوقُّف من فورنا ونقدِّم قائمة مرتَّبة في أي وقت يُطلب منَّا ذلك بلا سابق إنذار، تكون خوارزمية الترتيب بالإدراج هي الطريقة الملائمة.[2]

الترتيب بالجذر

لِنَعُد الآن إلى هوليريث. لم تُستخدم آلات الجدولة التي اخترعها خوارزميةَ الترتيب الانتقائي ولا خوارزميةَ الترتيب بالإدراج. بل استخدمت طريقةً سابقة لا تزال قيد الاستخدام حتى اليوم، تسمَّى «الترتيب بالجذر»، وتقديرًا لأول تطبيق للترتيب باستخدام الآلات، فإن الأمر يستحق أن نتوقَّف قليلًا عند طريقة الترتيب بالجذر للتعرُّف على آلية عملها. وتلك

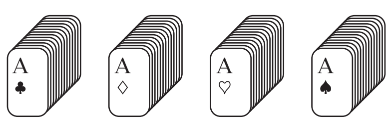
الطريقة مثيرة للاهتمام أيضًا؛ نظرًا لخلوها من أيِّ وجهِ مقارنةٍ بين العناصر المراد ترتيبها بواسطتها. على الأقل ليس بصورة كاملة كما سنرى. إضافة إلى ذلك، فإن طريقة الترتيب بالجذر ليست مهمة من المنظور التاريخي فحسب، بل إن أداءها بالغُ الروعة. فما الذي لا يُحَب في خوارزمية رائعة وعملية؟[3]

أسهل طريقة للتعرُّف على خوارزمية الترتيب بالجذر هي استخدام أوراق اللعب مرة أخرى. لنفترض أن لدينا مجموعةً كاملة مختلطة من أوراق اللعب نريد ترتيبها. توجد طريقة لذلك وهي تكوين ١٣ مجموعة، واحدة لكل قيمة مَنزَلة. نتفحَّص المجموعة ونأخذ كل ورقة ونضعها في المجموعة التي تنتمي لها. سنحصل على ١٣ مجموعة كلٌّ منها تتكوَّن من ٤ أوراق: مجموعة تتضمَّن كل أوراق الآس وأخرى تتضمَّن كل بطاقات العدد اثنين وهكذا.



ثم نجمع الأوراق مجموعةً تلو الأخرى ونتنبه إلى وضع كل مجموعة نأخذها تحت الأوراق التي نجمعها. بهذه الطريقة، ستكون كل الأوراق في أيدينا مرتَّبة جزئيًّا. ستكون أولُ أربع أوراق هي الآس، والأربع التي تليها هي الاثنين، وهكذا وصولًا إلى أوراق الملِك.

نكوِّن الآن أربع مجموعات جديدة، مجموعة لكل رمز. نتفحَّص الأوراق ونأخذ كل ورقة ونضعها في المجموعة التي تنتمي لها. سنحصل على أربع مجموعات من الرموز. ونظرًا لأن القيم قد رُتِّبت بالفعل، فسيكون لدينا في كل مجموعة كل البطاقات ذات الرمز الواحد مرتَّبة حسب القيمة.



للانتهاء من ترتيب الأوراق، لا نحتاج إلا إلى جمعها مجموعةً تلو الأخرى.

هذا هو جوهر طريقة الترتيب بالجذر. لم نرتِّب البطاقات بالمقارنة بينها جميعًا. بل أجرينا مقارناتٍ جزئية، حسب القيمة في البداية، ثم حسب الرمز.

بالطبع لو كان الترتيب بالجذر لا ينطبق إلا على أوراق اللعب، لما استحق اهتمامنا في هذا الكتاب. يمكن أن نرى كيف تتعامل خوارزمية الترتيب بالجذر مع الأعداد الصحيحة. لنفترض أن لدينا المجموعة التالية من الأعداد الصحيحة:

496	5	97	577	845	53	274	590	840	981	686
165	970	412	417	855	245	317	568	812	709	787
926	742	151	612	961	162	261	760	639	532	364

نتأكد من أن جميع الأعداد الصحيحة تتكوَّن من عدد الحدود نفسه. ومِن ثَم نضيف إلى الأعداد أصفارًا جهة اليسار إذا لزم الأمر بحيث نحوِّل العدد ٥ إلى ٠٠٥ والعدد ٩٧ إلى ٠٩٧ والعدد ٥٣ إلى ٠٥٣. نراجع جميع الأعداد ونصنِّفها حسب الحد الموجود أقصى اليمين. نستخدم ذلك الحد لوضع الأعداد في عشر مجموعات:

		742							
		612			165		417		
760	151	162			855		317		
970	961	532			245	926	787		
590	261	412		364	005	496	097		639
840	981	812	053	274	845	686	577	568	709

خفَّفنا لون تظليل الأعداد للإشارة إلى أنها قد رُتِّبت جزئيًّا؛ وتتضمَّن كل مجموعة الأعداد المشتركة في نفس الحد جهة اليمين. تنتهي كل الأعداد في المجموعة الأولى بالرقم صفر، وفي المجموعة الثانية تنتهي بالرقم ١، حتى مجموعة الأعداد الأخيرة حيث تنتهي بالرقم ٩. الآن، نجمع المجموعات العشر، مع البدء من المجموعة الأولى جهة اليسار وإضافة

المجموعات جهة الأسفل (مع الانتباه إلى عدم خلط الأعداد بأي طريقة). ثم نعيد توزيعها على عشر مجموعات باستخدام الحد الثاني جهة اليمين، ومِن ثَم يصبح لدينا الترتيب التالي:

						760			
						961			
	612					261			
	412			840		162			
	812			742	151	364	970	981	590
005	417		532	245	053	165	274	686	496
709	317	926	639	845	855	568	577	787	097

هذه المرة، كل الأعداد في المجموعة الأولى تحتوي على الرقم صفر في الخانة الثانية جهة اليمين؛ بينما تحتوي المجموعة الثانية على الرقم واحد في الخانة الثانية جهة اليمين، وهكذا في بقية المجموعات الأخرى. في الوقت نفسه، تُرتَّب العناصر في كل مجموعة حسب الحد الأخير؛ لأن هذا ما فعلناه عندما جمعنا المجموعات في المرة الأولى.

ننتهي بتجميع المجموعات وإعادة توزيع الأعداد باستخدام الرقم الثالث من جهة اليمين هذه المرة:

					532		709	812	926
005	151	245		412	568	612	742	840	961
053	162	261	317	417	577	639	760	845	970
097	165	274	364	496	590	686	787	855	981

الآن، تبدأ العناصر في كل مجموعة بالحد نفسه وتُرتَّب حسب الحد الثاني، نتيجةً لعملية تكوين المجموعات السابقة، وحسب الحد الأخير، نتيجةً لعملية تكوين المجموعات الأولى. لإكمال ترتيب الأعداد، نجمع فقط المجموعات مرة أخيرة.

يمكن تطبيقُ الترتيب بالجذر مع الكلماتِ أو أي تسلسل من حروف الأبجدية الرقمية وكذا الأعداد الصحيحة. في علم الكمبيوتر، نطلِق على تسلسل حروف الأبجدية الرقمية والرموز «سلسلة». يمكن تطبيق الترتيب بالجذر مع السلاسل، التي يمكن أن تتكوَّن من أرقام، كما في المثال الذي تناولناه، ولكن قد تكون أي نوع من السلاسل. عدد المجموعات في سلاسل الترتيب الأبجدي الرقمي يساوي عدد الحروف المميزة التي تؤلِّف الأبجدية (على سبيل المثال، تتكوَّن من ٢٦ مجموعة في أبجدية اللغة الإنجليزية)، ولكن ستكون العمليات مطابقة تمامًا. السمة المميزة لطريقة الترتيب بالجذر أننا نتعامل مع السلاسل على أنها سلاسل أبجدية رقمية لا سلاسل أرقام، حتى عندما تكون هذه السلاسل مؤلَّفة بالكامل من أرقام. إذا راجعت الطريقة التي اتبعناها، فسترى أننا لم نهتم بقيم الأعداد، ولكننا في كل مرة نتعامل مع حدٍّ معيَّن من حدود العدد، بالطريقة نفسها التي كنا سنتبعها عن طريق استخراج الحروف من كلمةٍ ما متجهين من اليمين إلى اليسار. ولذلك يُطلق على الترتيب بالجذر في بعض الأحيان اسم «طريقة الترتيب بالسلسلة».

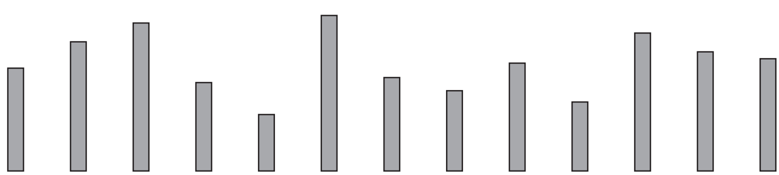
لا تدع تلك العبارة تخدعك وتجعلك تظن أن الترتيب بالجذر يمكن أن يرتِّب السلاسل بينما طرق الترتيب الأخرى التي تناولناها في هذا الكتاب لا تستطيع ذلك. كل الطرق يمكنها فعل ذلك. يمكننا ترتيب السلاسل ما دام يمكن ترتيب الرموز نفسها التي تتكوَّن منها. إن أسماء البشر هي سلاسل بالنسبة إلى الكمبيوتر، ومِن ثَم يمكننا ترتيبها لأن الحروف مرتَّبة أبجديًّا والأسماء يمكن مقارنتها معجميًّا. لقد جاء مسمَّى «الترتيب بالسلسلة» من كون الترتيب بالجذر يعامل كل المفاتيح، حتى الأرقام، باعتبارها سلاسل. أما طرق الترتيب الأخرى المذكورة في هذا الفصل فتُعامِل الأرقام كأرقام والسلاسل كسلاسل، وتعمل عن طريق مقارنة الأرقام أو السلاسل حسبما يقتضي الأمر. ونحن نستخدم الأرقام باعتبارها مفاتيح في الأمثلة التي نقدِّمها في مختلف خوارزميات الترتيب من باب التيسير ليس إلا.

تكمُن فاعلية طريقة الترتيب بالجذر في أنها تعالج العناصر المراد ترتيبها حدًّا بحدٍّ (أو حرفًا بحرف). فإذا كان لدينا n من العناصر نريد ترتيبها، وتتكوَّن العناصر من w من الحدود أو الحروف، فإن تعقيد الخوارزمية يساوي $O\left(wn\right)$. وهذا التعقيد أفضل بكثير من $O\left(n^2\right)$ المطلوب في الترتيب الانتقائي أو الترتيب بالإدراج.

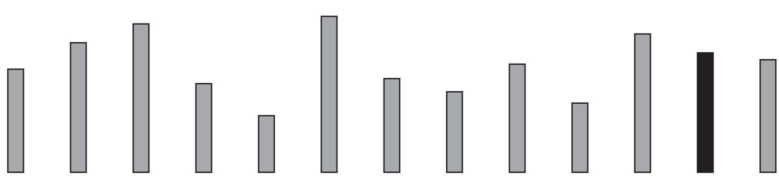
وها قد عُدنا إلى آلات الجدولة. كانت آلة الجدولة تعمل بطريقةٍ مشابهة لترتيب البطاقات المثقبة. تخيَّل أن لدينا مجموعة من البطاقات في كلٍّ منها عشرة أعمدة، وتشير الثقوب في كل عمود منها إلى رقم معيَّن. كانت الآلة تستطيع التعرُّف على الثقوب في كل عمود، وبذلك تتعرَّف على الرقم المطابق لها. وكان عامل التشغيل يضع البطاقات في الآلة، وتضع الآلة بدورها البطاقات في عشرة صناديق مخرجات اعتمادًا على العمود الأخير منها؛ أي أقل أرقامها أهميةً. كان عامل التشغيل يجمع البطاقات من صناديق المخرجات، مع الحرص على ألَّا يخلطها بأي حال، ويضعها مرة أخرى في الآلة وفي هذه المرة يوزِّعها في صناديق المخرجات باستخدام الرقم قبل الأخير؛ أي الرقم المجاور لأقل الأرقام أهمية. بعد تكرار هذه العملية عشر مرات، كان عامل التشغيل يستطيع جمع مجموعات من البطاقات المرتَّبة. وهذا هو المطلوب.

الترتيب السريع

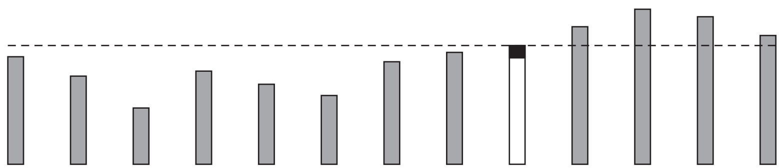
لنفترض أن لدينا مجموعة من الأطفال يلهون في ساحة ما (ربما في المدرسة) وتريد أن توقفهم صفًّا، من الأقصر إلى الأطول. في البداية، نطلب منهم الوقوف في صفٍّ وهو ما سيفعلونه بأي ترتيب يشاءون:



الآن، نختار طفلًا بصورة عشوائية:



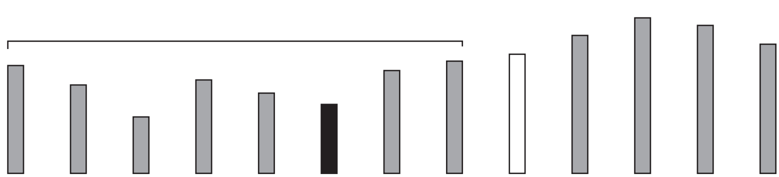
نخبر الأطفال أن يتنقلوا بحيث يتحرَّك كل الأطفال الأقصر من الطفل المختار جهة اليسار وباقي الأطفال جهة اليمين. في الشكل التالي، نوضِّح أين وقف الطفل المختار في النهاية، ويمكنك التحقُّق من أن الأطفال الأطول يقفون إلى يمينه والأقصر يقفون إلى يساره:



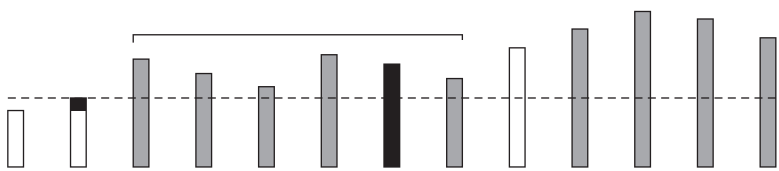
لم نطلب من الأطفال الوقوف بالترتيب الصحيح. لم نطلب منهم سوى التحرُّك نحو الطفل الذي اخترناه. ومِن ثَم شكلوا مجموعتين، إحداهما إلى يسار الطفل المختار والأخرى إلى يمينه. لا يقف الأطفال في هاتين المجموعتين بأي تسلسل من الأقصر إلى الأطول. لكننا نعلم يقينًا أن طفلًا «واحدًا» يقف في الموضع النهائي له في الصف الذي نحاول تكوينه؛ ذلك الطفل الذي اخترناه. كل الأطفال الواقفين إلى يساره أقصر منه، وكل الأطفال الواقفين إلى يمينه أطول منه. نطلق على الطفل الذي اخترناه «المحور»؛ لأن بقية الأطفال يتحركون حوله.

وكوسيلة مساعدة بصرية، نتَّبع طريقة تلوين الأطفال الذين يقفون في أماكنهم الصحيحة باللون الأبيض. وعندما نختار طفلًا كي يكون المحور، سنلوِّنه باللون الأسود؛ وعندما يتحرك بقية الأطفال حول المحور، سنستخدم قبَّعة سوداء صغيرة للإشارة إلى الموضع النهائي للمحور (الملوَّن بالأبيض؛ لأنه في المكان الصحيح ورأسه باللون الأسود للإشارة إلى أنه المحور).

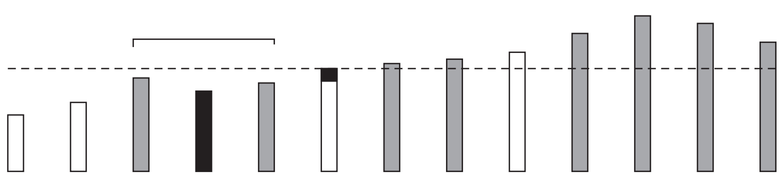
الآن، نحوِّل انتباهنا إلى مجموعة من المجموعتين — جهة اليمين أو اليسار — ولنقُل المجموعة جهة اليسار. مرة أخرى، نختار محورًا عشوائيًّا في تلك المجموعة:



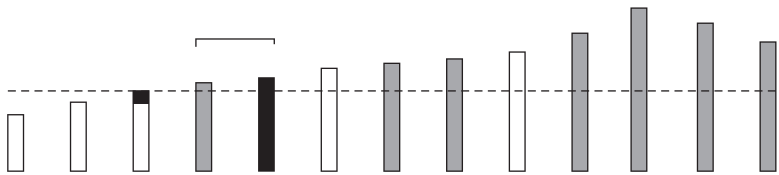
نطلب من الأطفال في تلك المجموعة أن يفعلوا الشيء نفسه الذي فعلوه من قبل: التحرُّك بحيث ينتقلون إلى يسار المحور، إذا كانوا أقصر، ونحو اليمين إذا كانوا أطول. مرة أخرى، سيصبح لدينا مجموعتان جديدتان أصغر، كما يمكنك أن ترى أدناه. إحدى المجموعتين مكوَّنة من طفل واحد؛ ومِن ثَم يقف الطفل في الموضع الصحيح في تلك المجموعة الصغيرة. ومِن ثَم يكون بقية الأطفال واقفين على يمين المحور الثاني. يقف المحور الثاني في المكان الصحيح، حيث يقف كل الأطفال الأقصر إلى يساره والباقون جميعًا إلى يمينه. المجموعة جهة اليمين تمتد إلى المحور الأول. عندها نختار محورًا جديدًا ثالثًا من تلك المجموعة.



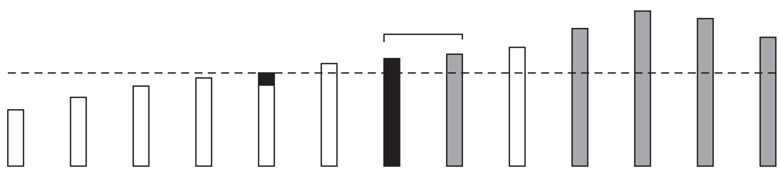
عندما نخبر الأطفال في المجموعة أن يتحركوا مثل المرة السابقة، حسب طولهم بالنسبة إلى المحور الثالث، ستتكوَّن مجموعتان صغيرتان. نركِّز على المجموعة جهة اليسار. ونفعل كما فعلنا في السابق. نختار محورًا — رابعًا — ونطلب من الأطفال في هذه المجموعة المكوَّنة من ثلاثة أن يتحركوا حول هذا المحور.



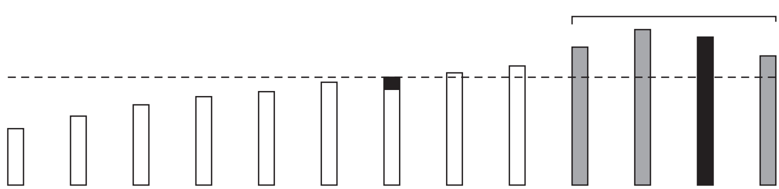
عندما يفعلون ذلك، ينتهي الأمر بالمحور إلى أن يصبح الأول بين الثلاثة، ومِن ثَم تتبقى لدينا مجموعة مكوَّنة من طفلين على يمين المحور. نختار واحدًا من الاثنين ليكون محورًا، وسيتحرك الطفل الآخر، إذا لزم الأمر، إلى يمين المحور.



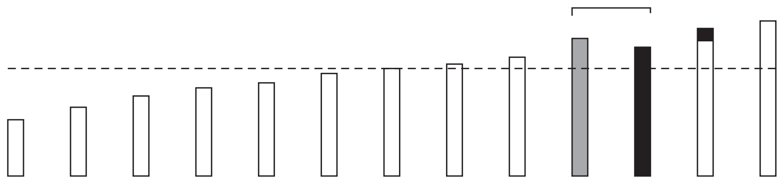
يتبيَّن أن هذا الطفل ليس بحاجة إلى الانتقال من مكانه على الإطلاق. وهكذا نكون قد تمكَّنا الآن من ترتيب نصف الأطفال تقريبًا؛ علمًا بأن هناك مجموعتين تركناهما عندما كنَّا نتعامل مع المحاور السابقة. نعود إلى المجموعة الأولى منهما من اليسار حتى نختار محورًا ونكرِّر العملية.



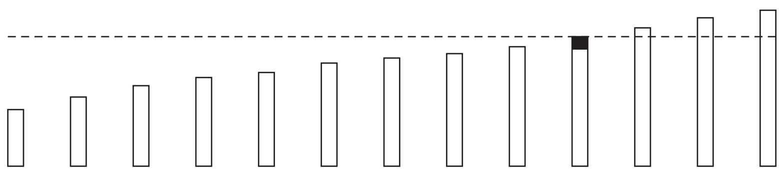
مرة أخرى، لا حاجة إلى التحرُّك من الموضع؛ ومِن ثَم ننتقل إلى المجموعة الأخيرة من الأطفال الذين لم يُرتَّبوا لنختار محورًا من بينهم.



يصبح لدينا مجموعة مكوَّنة من طفل واحد على يمين المحور، ومجموعة مكوَّنة من طفلين على يسار المحور. نركِّز على المجموعة جهة اليسار ونختار المحور الأخير من الطفلين.



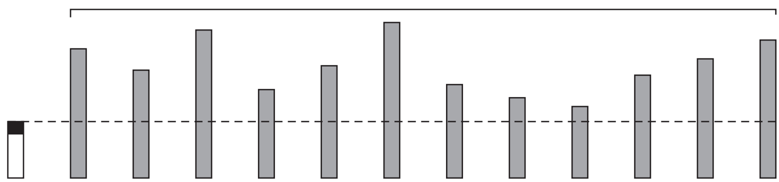
ها قد انتهينا. وصار الأطفال يقفون بترتيب حسب طول القامة.



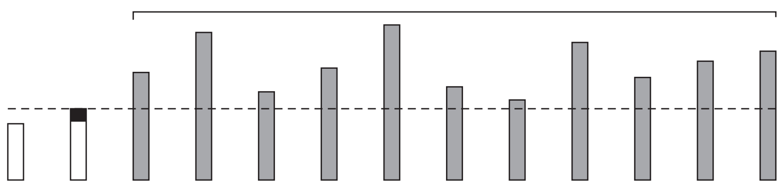
لنوضِّح ما قمنا به. لقد تمكَّنا من ترتيب الأطفال عن طريق وضع طفل واحد في موضعه الصحيح في كل مرة. وللقيام بذلك، لم نحتَج إلا إلى مطالبة بقية الأطفال بأن يتحركوا حول ذلك الطفل. بالطبع سوف تؤتي هذه الطريقة ثمارها دائمًا، ليس مع الأطفال فحسب، بل مع أي شيء نريد ترتيبه. فإذا كان لدينا مجموعة من الأعداد يمكن ترتيبها، يمكننا اتِّباع عملية مماثلة، وذلك باختيارِ عددٍ ما عشوائيًّا، وننقل بقية الأعداد حوله بحيث ينتهي المآل بالأعداد الأصغر قبل الرقم المختار والأعداد الأكبر بعده. سنكرِّر العملية في المجموعات الأصغر التي تكوَّنت؛ وفي النهاية، سنجد كل الأعداد في الترتيب الصحيح. وهذه العملية هي أساس خوارزمية «الترتيب السريع».

يعتمد الترتيب السريع على ملاحظة أنه إذا أمكن وضع عنصر واحد في الموضع الصحيح بالنسبة إلى بقية العناصر — بغض النظر عن مكان هذا الموضع — ثم تكرار العملية مع بقية العناصر، فسينتهي الأمر بوضع جميع العناصر في مواضعها الصحيحة. إذا عُدنا بالذاكرة إلى ما قمنا به في عملية الترتيب الانتقائي، فسنجد أننا أخذنا كل عنصر من العناصر أيضًا ووضعناه في المكان الصحيح بالنسبة إلى بقية العناصر، ولكن العنصر الذي أخذناه كان العنصر الأصغر دومًا من بين العناصر المتبقية. وهذا فرقٌ بالغ الأهمية؛ ففي الترتيب السريع، «لسنا» بحاجةٍ إلى اختيار العنصر الأصغر من بين العناصر المتبقية ليكون المحور. لِنرَ ما سيحدث إذا قمنا بذلك.

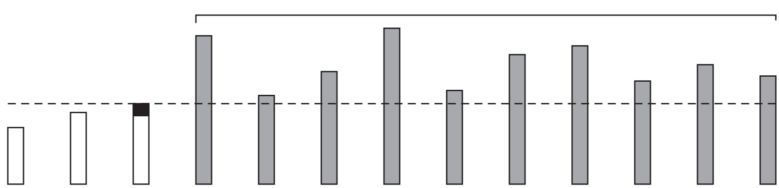
إذا بدأنا مرةً أخرى بمجموعة الأطفال نفسها، فسنجعل أقصر طفل فيهم هو المحور. سينتقل هذا الطفل إلى بداية الصف، وسيتحرك باقي الأطفال خلف المحور.



بعد ذلك، سنختار الطفل الأطول من الطفل الأول مباشرةً ونضعه ثانيًا في الصف. وسينتقل باقي الأطفال مرة أخرى خلف المحور.



وبالقيام بالأمر نفسه مع الطفل الثالث، نحصل على الصورة التالية:



لكن لاحظ مدى التشابه الغريب لهذا مع الفرز الانتقائي؛ إذ إننا نرتِّب الصف من اليسار إلى اليمين بدءًا من الطفل الأقصر بين الأطفال المتبقين.

لم نذكر كيف اخترنا عنصرًا في كل مرة كي يكون المحور. نرى الآن أننا لسنا مضطرين إلى اختيار العنصر الأصغر من بين العناصر. أولًا: لأن اختيار العنصر الأصغر يتطلب جهدًا؛ إذ علينا أن نبحث في كل مرة عن العنصر الأصغر ونجده. ثانيًا: تلك العملية

تسير على نهج خوارزمية نعرفها بالفعل، ومِن ثَم لا يُفترض أن تكون هناك فائدة كبيرة تُرجى من القيام بذلك.

الحقيقة أن الترتيب السريع أفضلُ من الترتيب الانتقائي؛ لأننا «بطبيعة الحال» (وسنعرف المقصود بكلمة بطبيعة الحال بعد قليل) سوف نختار محورًا يقسم البيانات التي لدينا بطريقةٍ أكثر تساويًا. أما اختيار العنصر الأصغر فيؤدي إلى قسمة غير متساوية إلى أقصى حدٍّ؛ إذ لا توجد عناصر على يسار المحور، وكل العناصر المتبقية توجد على يمين المحور. ومِن ثَم لا نتمكَّن إلا من تحديد موضع المحور نفسه في كل مرة.

أما إذا كانت القسمة متساوية أكثر، فإننا لا نتمكَّن من تحديد موضع المحور فحسب. بل نتمكَّن أيضًا من تحديد المواضع الصحيحة لجميع العناصر على يسار المحور «بالنسبة إلى العناصر الواقعة على يمين المحور». صحيح أن تلك العناصر ليست في مواضعها النهائية بعد. ولكنها بوجه عام في مواضع أفضل من ذي قبل. وهكذا يكون لدينا عنصر واحد — المحور — في أفضل موضع ممكن، والعناصر الأخرى في مواضع أفضل من ذي قبل.

ولهذا الأمر تأثيرٌ مهم في أداء خوارزمية الترتيب السريع؛ فتعقيد الخوارزمية المتوقَّع يساوي $O\left(nlgn\right)$ وهو أفضل بكثير من التعقيد $O\left(n^2\right)$. فإذا أردنا ترتيب مليون عنصر، تتحوَّل $O\left(n^2\right)$ إلى ١٠^١٢ أي تريليون، ولكن $O\left(nlgn\right)$ تساوي نحو ٢٠ مليونًا.

يعتمد الأمر برُمَّته على اختيار المحور المناسب. فليس من المنطق أن نبحث في كل مرة عن محورٍ يقسم البيانات بأفضل طريقة ممكنة؛ إذ سيتطلب الأمر البحث لإيجاد المحور المناسب، ما يضيف مزيدًا من التعقيد إلى العملية. إذن، فالاستراتيجية الجيدة هي ترْك الأمر إلى الحظ. ما علينا سوى اختيار محور عشوائي واستخدامه لتقسيم البيانات.

كي نعرف السببَ وراء صلاحية هذه الاستراتيجية، لِنرَ أولًا الأسباب التي تجعلها غير سيئة. ستكون استراتيجية سيئة لو أدَّت إلى سلوك يشبه السلوك الذي رأيناه لتوِّنا لما تحوَّل الترتيب السريع إلى ترتيب انتقائي. وهو ما قد يحدث إذا انتقينا المحور في كل مرة من عنصر لا يقسم العناصر بالتساوي فعليًّا. ويمكن أن يحدث ذلك الأمر إذا انتقينا في كل مرة العنصر الأصغر أو الأكبر من بين العناصر (الموقف سيان). ويمكن أن تبلغ الاحتمالية الإجمالية لحدوث كل هذا $2^{n-1}/n!$.

يصعب استيعاب احتمالية بقيمة $1/n!$؛ لأنها منخفضة إلى أقصى الحدود. وكي نوضِّحها بالسياق، إذا أخذت مجموعة مكوَّنة من ٥٢ ورقة لعب وخلطتها عشوائيًّا، فإن احتمالية أن تصبح المجموعة مرتَّبة في النهاية تساوي ١ / ٥٢! هذا الأمر مشابه تقريبًا

لقذف عملة معدنية وإنزالها على الصورة ٢٢٦ مرة على التوالي. وعندما تضرب في 2^{n-1}، لا تتحسَّن الأمور كثيرًا. فالعدد !٥١٢ / ٥٢ يساوي تقريبًا ٢٫٨ × ١٠^-٥٣^. ولوضع المسألة في منظور كونيٍّ، تتكوَّن الأرض من ١٠^٥٠^ ذرة تقريبًا. إذا كان عليك أن تنتقي أنت وأحد أصدقائك ذرةً من الأرض كلٌّ بمفرده، فإن احتمالية أن تنتقيا الذرة نفسها تساوي ١٠^-٥٠^، وتلك القيمة فعليًّا أكبر من !٥١٢ / ٥٢؛ وهي احتمالية الترتيب السريع غير العادية بشأن مجموعة بطاقات اللعب.[4]

بذلك يتضح أننا «بطبيعة الحال» نختار محورًا يقسم المسألة بطريقة أكثر تساويًا، كما ذكرنا من قبل. وباستثناء سلسلة الحظ السيئ بشأن النسب الكونية، فإننا لا نتوقَّع أن نختار أسوأ محور ممكن في كل مرة. فالاحتمالات، في واقع الأمر، تصب في صالحنا أكثر؛ فباختيار المحاور عشوائيًّا، نتوقَّع أن تكون قيمة التعقيد $O\ (nlgn)$. من الناحية النظرية، يُحتمل أن يكون التعقيد أسوأ من ذلك، ولكن أهمية الاحتمالية هي أهمية أكاديمية فحسب. وستعمل خوارزمية الترتيب السريع بالسرعة التي نتوقَّعها لجميع الأغراض العملية.

وُضعت خوارزمية الفرز السريع على يد عالِم الكمبيوتر البريطاني توني هور بين عامي ١٩٥٩–١٩٦٠.[5] ربما تكون الخوارزمية الأكثر شهرةً وانتشارًا بين خوارزميات الفرز والترتيب في الوقت الحاضر؛ لأنها تتفوق على كل الخوارزميات الأخرى عند تنفيذها بالطريقة الصحيحة. كما أنها أول خوارزمية نراها سلوكها ليس حتميًّا بالكامل. فعلى الرغم من أنها ستقوم بعملية الترتيب بالطريقة الصحيحة على الدوام، فلا يمكننا أن نضمن أنها ستستغرق مدة التنفيذ نفسها دومًا. يمكننا أن نضمن الاستبعاد التام لفكرة إظهارها سلوكًا غيرَ عاديٍّ. وهذا مفهوم مهم لأنه يقودنا إلى ما يسمَّى ﺑ «الخوارزميات العشوائية»؛ وهي تلك الخوارزميات التي تستخدم عنصر المصادفة في تشغيلها. وهذا يتناقض مع حدْسنا؛ إذ نتوقَّع أن تكون الخوارزميات هي تلك الخوارزميات القاطعة ذات السلوك المتوقَّع، التي تتَّبع التعليمات التي نضعها لها على مسارٍ محدَّد سلفًا وهي صاغرة. ولكن ازدهرت الخوارزميات العشوائية في السنوات الأخيرة؛ إذ تبيَّن أن المصادفة يمكن أن تساعدنا في حل المسائل التي لا تزال مستعصية الحل على الأساليب الأكثر نمطية.[6]

الترتيب بالدَّمج

تعرَّفنا على خوارزمية الترتيب بالجذر التي ترتِّب العناصر بالأساس عن طريق التوزيع؛ حيث تضع كلَّ عنصر في مجموعته الصحيحة في كل دورة ترتيب للبيانات. والآن، سنتناول

طريقةَ ترتيبٍ أخرى ترتِّب العناصر عن طريق «دمجها» معًا بدلًا من تقسيمها. الطريقة تسمَّى «الترتيب بالدَّمج».

تبدأ خوارزمية الترتيب بالدَّمج بالتسليم بقدرتها المحدودة على الترتيب والتصنيف؛ تخيَّل أننا لا نستطيع ترتيب العناصر إذا أُعطيت لنا بأي ترتيب عشوائي. ليس بمقدورنا سوى القيام بالتالي: إذا كان لدينا مجموعتان من العناصر، وكل مجموعة مرتَّبة بالفعل، يمكننا دمجهما معًا والحصول على مجموعة واحدة مرتَّبة.

> ازدهرت الخوارزميات العشوائية في السنوات الأخيرة؛ إذ تبيَّن أن المصادفة يمكن أن تساعدنا في حل المسائل التي لا تزال مستعصية الحل على الأساليب الأكثر نمطية.

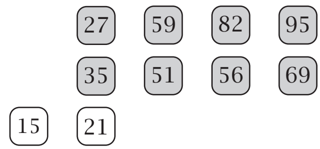
على سبيل المثال، لنقُل إن لدينا المجموعتين التاليتين، واحدة في كل صف (على الرغم من أن المجموعتين في المثال لهما العدد نفسه من العناصر، فلا يلزم أن تكون المجموعتان بالحجم نفسه):

15	27	59	82	95
21	35	51	56	69

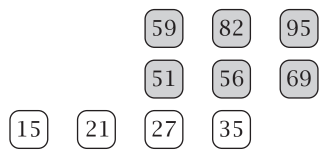
كما ترى، كل مجموعة من المجموعتين مرتَّبة بالفعل. نريد دمج المجموعتين من أجل الحصول على مجموعةٍ واحدة مرتَّبة. وهذه عملية بسيطة للغاية. نتحقَّق من العنصر الأول في كل مجموعة. فنجد أن ١٥ أصغر من ٢١، ومِن ثَم سيكون هذا هو العنصر الأول في المجموعة الثالثة التي نعمل على تكوينها:

	27	59	82	95
21	35	51	56	69
15				

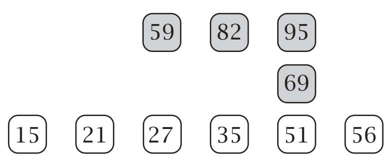
نتحقَّق مرة أخرى من العناصر الأولى في المجموعتين، وفي هذه المرة، نرى أن العدد ٢١ في المجموعة الثانية أصغر من العدد ٢٧ في المجموعة الأولى. ومِن ثَم نأخذ العدد ونُلحِقه بالمجموعة الثالثة.



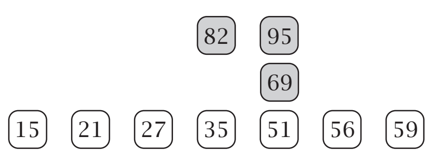
إذا واصلنا على هذه الوتيرة، فسنأخذ العدد ٢٧ من المجموعة الأولى، ثم ٣٥ من المجموعة الثانية ونضيفهما إلى نهاية المجموعة الثالثة كما يلي:



الآن، ٥١ أصغر من ٥٩، وكذلك ٥٦ أصغر من ٥٩. وبما أننا بالفعل نقلنا العدد ٣٥ من المجموعة الثانية إلى الثالثة، نكون في النهاية قد نقلنا ثلاثة عناصر متتالية من المجموعة الثانية إلى الثالثة. وهذا شيء رائع؛ لأننا بذلك نحافظ على العناصر في المجموعة الثالثة مرتَّبة. فما من سببٍ يستدعي تقليل حجم أول مجموعتين بالمعدَّل نفسه.



نعود إلى المجموعة الأولى، حيث العدد ٥٩ أصغر من ٦٩، ومِن ثَم نضيفه إلى المجموعة الثالثة:



بعد ذلك، وبنقل العدد ٦٩ إلى المجموعة الثالثة، نكون قد أفرغنا المجموعة الثانية بالكامل:

82 95

15 21 27 35 51 56 59 69

ننهي العملية بنقل آخر العناصر المتبقية من المجموعة الأولى إلى المجموعة الثالثة، التي هي قطعًا أكبرُ من العنصر الأخير في المجموعة الثالثة، وإلَّا لما نقلناه هناك أولًا. ها قد صارت جميع العناصر مرتَّبة الآن:

15 21 27 35 51 56 59 69 82 95

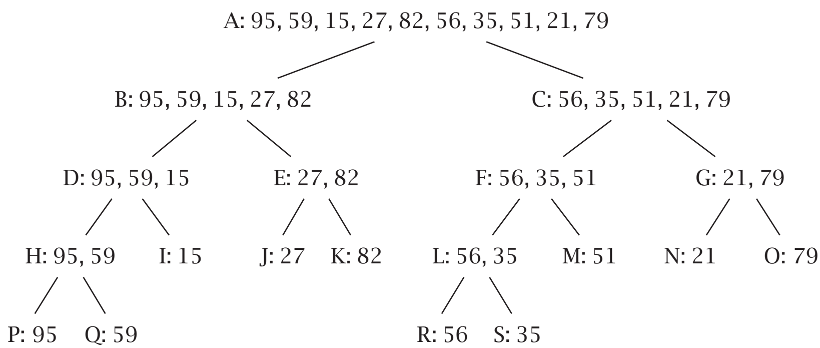
جميل أن تكون هناك طريقة لإخراج مجموعة مرتَّبة من مجموعتين مرتَّبتين، ولكن لا يبدو أن تلك الطريقة تقدِّم حلًّا لمسألة ترتيب مجموعة واحدة ذات عناصر غير مرتَّبة. صحيح أنها لا تقدِّم حلًّا، ولكنها عنصر مهم من عناصر الحل.

تخيَّل الآن أن لدينا مجموعة من الأشخاص. أعطينا واحدًا منهم مجموعة عناصر كي يرتِّبها. هذا الشخص لا يعرف طريقة الترتيب، ولكنه يعلم جيدًا أنه إذا كان لديه جزءان من هذه العناصر مرتَّبان بصورة أو بأخرى، يمكنه أن يُخرِج منهما مجموعةً مرتَّبة في النهاية. ومِن ثَم سيفعل ذلك الشخص ما يلي: سيقسم المجموعة إلى جزأين ويعطيهما لشخصين آخرين. يقول للأول منهما: «خذ تلك المجموعة ورتِّبها. وبمجرد أن تنتهي، أعِدْها إليَّ.» ويقول الشيءَ نفسه للشخص الثاني. ثم ينتظر.

إن الشخص الأول لا يعرف كيف يرتِّب العناصر، ولكن إذا نجح الشخصان الآخران بصورةٍ ما في ترتيب الجزأين اللذين معهما وأعاداهما إليه، فسيعيد لنا الشخص الأول المجموعة النهائية المرتَّبة بالكامل. لكن الشخصين الآخرين لا يعرفان أكثرَ مما يعرفه الشخص الأول — إنهما لا يعرفان كيفية الترتيب، بل فقط يعرفان طريقة دمج عناصر مرتَّبة باستخدام الخوارزمية الواردة أعلاه — إذن هل تُوصِّل إلى أي شيء؟

الإجابة نعم، بشرط أن يفعل كلٌّ منهما ما قام به الأول: يقسم كلٌّ واحد منهما الجزء الذي معه إلى جزأين، ويعطي كلٌّ منهما الجزأين اللذين معه إلى شخصين آخرين، ثم ينتظرانهما حتى يفعل كلُّ شخصٍ ما أُسند إليه ويعطيانه الجزأين وقد رُتِّبا.

تشبه تلك الطريقة لعبة تمرير المسئولية، ولكن انظر ما يحدث إذا حاولنا توضيح المسألة بالمثال. نبدأ بالأعداد ٩٥ و٥٩ و١٥ و٢٧ و٨٢ و٥٦ و٣٥ و٥١ و٢١ و٧٩. نعطي تلك الأعداد إلى أليس (A) التي تقسمها بدورها إلى جزأين وتعطيهما إلى بوب (B) وكارول (C). يمكنك أن ترى ذلك في المستوى الأول من الشجرة المقلوبة فيما يلي:

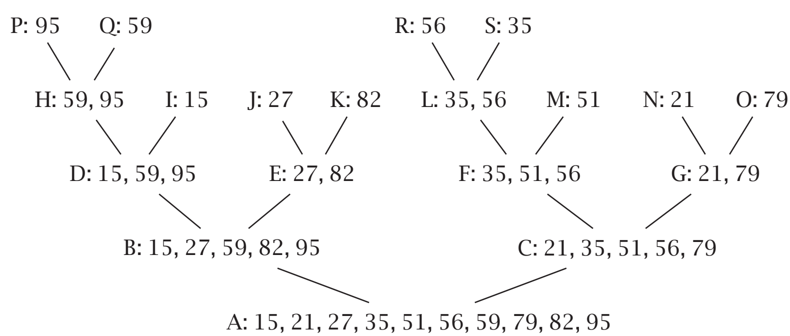


ثم يقسم بوب الأعداد التي معه إلى جزأين ويمرِّرهما إلى ديف (D) وإيف (E). وبالمثل، تقسم كارول الأعداد التي معها وتمرِّرها إلى فرانك (F) وجريس (G). تستمر مجموعة الشخصيات في تمرير المسئولية. فيُقسِّم ديف الأعداد التي معه على هيدي (H) وإيفان (I)؛ وتوزِّع إيف العددين اللذين معها على جودي (J) وكارين (K)، بينما يقسم فرانك الأعداد على ليو (L) ومالوري (M)، وتقسم جريس الأعداد على نيك (N) وأوليفيا (O). في النهاية، تقسم هيدي جزأيها على بيجي (P) وكوينتين (Q)، ويقسم ليو جزأيه على روبرت (R) وسيبيل (S).

الأفراد عند مستوى أوراق الشجرة ليس لديهم ما يفعلونه في الحقيقة. فكلٌّ من بيجي وكوينتين يتلقيان عددًا واحدًا ويُطلب منهما ترتيبه. ولكن عددًا واحدًا لا يحتاج إلى الترتيب بطبيعة الحال؛ فهو مرتَّب بذاته. ومِن ثَم يعيد بيجي وكوينتين العددين إلى هيدي. وكذلك يعيد إيفان وجودي وكارين وسيبيل ومالوري ونيك وأوليفيا الأعداد التي تلقَّوها.

لننتقل الآن إلى الشجرة التالية. في هذه الشجرة، سننتقل من الأوراق في القمة (ولهذا تبدو هذه الشجرة عادية وليست مقلوبة) إلى الجذور في الأسفل. لنركِّز على هيدي. تستعيد هيدي عددين، رُتِّب كلٌّ منهما (بلا أي قيمة تُذكر). تعرف هيدي كيف تَدمُج مجموعتين مرتَّبتين لإنشاء مجموعةٍ واحدة، ومِن ثَم تستطيع استخدام العددين ٩٥ و٥٩ لتُنشِئ المجموعةَ ٥٩ و٩٥. بعد ذلك تعيد تلك المجموعة المرتَّبة المكوَّنة من عددين إلى ديف. سيفعل ليو الأمر نفسه: سيحصل على العددين ٣٥ و٥٦ المرتَّبين بالفعل (تلقائيًّا) ويعلم كيف يرتِّب هذين العددين ويُنشِئ المجموعةَ ٣٥ و٥٦ ويعيدها إلى فرانك.

لا يملك ديف أدنى فكرة عن الأعداد ٩٥ و٥٩ و١٥ التي تلقاها في البداية؛ إذ تلقَّى ٥٩ و٩٥ من هيدي و١٥ من إيفان. كلتا هاتين المجموعتين مرتَّبة بالفعل، ما يعني أن ديف يمكنه دمجهما وتكوين المجموعة ١٥ و٥٩ و٩٥. وبالطريقة نفسها، يحصل فرانك على العددين ٣٥ و٥٦ من ليو وعلى العدد ٥١ من مالوري ويمكنه تكوين المجموعة ٣٥ و٥١ و٥٦.



إذا تصرَّف كلُّ فرد بالطريقة نفسها، فسيصبح لدى أليس، عندما تصلها الأعداد، قائمتان مرتَّبتان، إحداهما من كارول والأخرى من بوب. عندئذٍ ستدمج القائمتين لإنشاء القائمة النهائية المرتَّبة.

هاتان الشجرتان هما جوهر الترتيب بالدَّمج. نسند عملية الترتيب إلى أكبر عدد ممكن حتى لا يمكن إجراء الترتيب؛ لأن العناصر المفردة عناصر مرتَّبة بطبيعة الحال. ثم ندمج مجموعاتٍ أكبر وأكبر إلى أن نستوعب كل العناصر في مجموعة واحدة نهائية مرتَّبة.

مقدار الذكاء المطلوب من الشخصيات قليل جدًّا. يمكنك أن ترى في الشجرة الأولى أن إيف تلقَّت من بوب مجموعةَ أعداد تصادف أنها مرتَّبة بالفعل: ٢٧ و٨٢. هذا لا يهم.

فهي لا تتوقَّف عن التحقُّق ممَّا إذا كانت المجموعة بحاجة إلى ترتيب من عدمه، ونحن لا نريدها أن تفعل ذلك؛ لأن مثل هذا التحقُّق سيستغرق وقتًا. كل ما تفعله هو تقسيم العناصر وتمريرها إلى شخصٍ آخر. وستستعيدها وتدمجها لإنشاء المجموعة التي كانت معها بالفعل. لا بأس؛ في المخطَّط الأكبر للعناصر، لن يؤثِّر هذا التعاون غير المبرَّر بين إيف وجودي وكارين في أداء الخوارزمية.

تعقيد خوارزمية الترتيب بالدمج جيدٌ مثل خوارزمية الترتيب السريع؛ إذ يساوي $O\ (nlgn)$. وهذا يعني أن لدينا خوارزميتين لهما الأداء نفسه. وفي الجانب العملي، قد يختار المبرمجون إحداهما أو الأخرى بناءً على عوامل أخرى إضافية. وعادةً ما تكون برامج الترتيب السريع أسرع من برامج الترتيب بالدمج نظرًا لسرعة تنفيذها الفعلي في لغة البرمجة. أما الترتيب بالدمج فيقسِّم البيانات قبل دمجها، ما يعني أنه يمكن تشغيلها في عمليات متوازية، بحيث يمكن ترتيب كميات هائلة من البيانات باستخدام مجموعة من أجهزة الكمبيوتر، حيث يعمل كل جهاز مثل الشخصيات في مثال الترتيب الموضَّح آنفًا.

تعتبر خوارزمية الترتيب بالدمج قديمةً قِدَم الكمبيوتر. كان مخترعها هو الأمريكي من أصول مجرية نيومان يانوس لايوس المعروف باسمه الأمريكي جون فون نيومان (١٩٠٣–١٩٥٧). في عام ١٩٤٥، كتب مخطوطة بالحبر في ٢٣ صفحة لواحد من أوائل أجهزة الكمبيوتر الرقمية وهو الكمبيوتر التلقائي المنفصل المتغير أو EDVAC اختصارًا. في أعلى الصفحة الأولى، كُتبت عبارة «سري للغاية» بالقلم الرصاص (ثم مُحيت فيما بعد)؛ لأن العمل على أجهزة الكمبيوتر في عام ١٩٤٥ كان من الأعمال السرِّية المحظور الاطلاع عليها نظرًا لصلتها بالجيش. كان موضوع المخطوطة تطبيقًا غير رقمي لأجهزة الكمبيوتر وهو: الترتيب. والطريقة التي وصفها فون نيومان في هذه المخطوطة هي ما نسميه الآن الترتيب بالدَّمج.[7]

الفصل الخامس

خوارزمية بيج رانك

إذا كنت دون عمرٍ معيَّن، فالكلمات هوت بوت ولايكوس وإكسايت وألتا فيستا وإنفوسيك لن تعني لك شيئًا، أو إن كانت تعني شيئًا، فربما لن يكون معناها محرِّكات بحْث. غير أن جميع تلك المحركات كانت تتبارى من أجل جذب اهتمامنا في وقتٍ ما، سعيًا إلى جذبنا لاستخدامها بوابة إلى الشبكة العنكبوتية.

لقد أصبح ذلك الآن شيئًا من الماضي؛ إذ يسيطر على المشهد في مجال محركات البحث خدمتان، وهما محرك جوجل الذي تديره شركة ألفابت، ومحرك بينج الذي تديره شركة مايكروسوفت. إن ظهور العديد من الحلول المتنافسة في سوقٍ جديدة على نطاق واسع، ثم اندماج تلك الحلول في وقتٍ لاحق، يُعَد نمطًا من الأنماط التي شهدناها في العديد من الصناعات على مرِّ التاريخ. واللافت للنظر في مجال محركات البحث هو معرفتنا بأن النجاح الهائل الذي حقَّقه محرك البحث جوجل له عامل كبير في هذا التطور، وهو النجاح القائم بدوره على خوارزميةٍ اخترعها مؤسسو محرك البحث. كان المؤسسان هما لاري بيج وسيرجي برين — طالبا الدكتوراه بجامعة ستانفورد — وأطلقا على هذه الخوارزمية بيج رانك (Page Rank)، نسبةً إلى مخترعها بيج (وليس اشتقاقًا من اسم لفظة page بمعنى «صفحة» ولفظة rank بمعنى «تصنيف» كما قد تتوقَّع).

قبل أن نشرع في وصف خوارزمية بيج رانك، ينبغي أن نفهم تحديدًا ما تفعله محركات البحث. إنها تقوم بوظيفتين في الحقيقة. أولًا: تتسلَّل إلى الشبكة العنكبوتية وتقرأ كل صفحات الويب التي يمكنها الوصول إليها وتُفهرِسها. بهذه الطريقة، عندما نكتب شيئًا في خانة البحث، تبحث محركات البحث في البيانات التي خزَّنتها على صفحات الويب التي تسلَّلت إليها وتعثر على البيانات التي تتطابق مع استفسارنا. لذا إذا بحثنا

عن «تغيُّر المناخ»، فستبحث محركات البحث في البيانات التي جمعتها لإيجاد صفحات الويب التي تحتوي على كلمات البحث.

إذا كانت كلمة البحث المستخدمة تصف موضوعًا شائعًا، يمكن أن نخرج بعدد هائل من النتائج. في وقت تأليف هذا الكتاب، يعود السؤال عن «تغيُّر المناخ» على جوجل بما يزيد على ٧٠٠ مليون نتيجة؛ قد يختلف هذا الرقم عندما تقرأ تلك السطور، ولكن على الأقل لديك لمحة عن نطاق النتائج. وهذا يقودنا إلى الوظيفة الثانية التي تقوم بها محركات البحث. لا بد أن تعرض لنا نتائج البحث بحيث تظهر النتائج الأكثر ارتباطًا بما نبحث عنه في البداية، وتظهر النتائج التي لا يُحتمل أنها تهمنا لاحقًا. فإذا كنت تسعى إلى معرفة الحقائق عن تغير المناخ، فستتوقع أن ترى النتائج من مواقع الأمم المتحدة أو الإدارة الوطنية للملاحة الجوية والفضاء (ناسا) أو ويكيبيديا في صدارة النتائج الظاهرة لك. وستتفاجأ إلى حدٍّ ما إذا تصدرت نتائجَ البحث صفحةُ ويب تشرح وجهةَ نظر «جمعية الأرض المسطحة» عن الموضوع. ومن بين مئات الملايين من صفحات الويب التي قد تتعلق باستفسارك، سيكون العديد منها بلا قيمة؛ وقد تتسم صفحات أخرى بالسطحية والثرثرة، ولكن ثمة صفحات أخرى لن يرجى منها فائدة على الإطلاق. أنت تريد أن تركِّز على تلك الصفحات الموثوق فيها التي تتحدَّث في صلب الموضوع.

عندما ظهر محرك البحث جوجل على الساحة (المؤلِّف في سنٍّ تسمح له بتذكُّر ذلك)، بدأ الناس (ومن بينهم المؤلِّف) في التحوُّل من محركات البحث القديمة المنقرضة في الوقت الحالي إلى محرك البحث الجديد؛ لأن نتائجه كانت أفضل وكانت تصل أسرع. وكان من العوامل المفيدة أيضًا بساطة صفحة الويب لجوجل؛ إذ كانت لا تحتوي إلا على المعلومات ذات الصلة بدلًا من تدفُّق أنواع الأدوات كافة في وجهك؛ إذ كان ذلك نمط الصفحات السائد حينذاك. سنُنحِّي العامل الثاني جانبًا على الرغم من أهميته (فقد أدركت جوجل أن المستخدمين يهتمون بنتائج البحث الجيدة والسريعة لا بالأجراس والصافرات)، وسنتناول العامل الأول. كيف تمكَّنت جوجل من تقديم نتائج أفضل وأسرع من محركات البحث الأخرى؟

لو كانت الشبكة العنكبوتية صغيرة، لتمكَّنا من إنشاء كتالوج لها، ولأصبح لدينا محررون لتنظيم ذلك الكتالوج ويعينون درجات الأهمية لمدخلاته؛ أي صفحات الويب. ولكن حجم الشبكة العنكبوتية يحول دون اتباع مثل هذا النهج، على الرغم من وجود محاولات سابقة في هذا الصدد، قبل أن يتضح أن هذا الحجم الهائل للشبكة سيجعل من تنفيذ هذه المهمة أمرًا مستحيلًا.

إذا كنتَ تسعى إلى معرفة الحقائق عن تغيُّر المناخ … فستتفاجأ إلى حدٍّ ما إذا تصدَّرت نتائج البحث صفحة ويب تشرح وجهة نظر «جمعية الأرض المسطحة» عن الموضوع.

تتألَّف الشبكة العنكبوتية من صفحات ويب يرتبط بعضها ببعض من خلال روابط. نطلق على تلك الروابط «الارتباطات التشعُّبية»، والنص الذي يحتوي على مثل هذه الإسنادات الترافقية إلى أجزاءٍ أخرى من النص أو النصوص الأخرى يُطلق عليه «النص التشعبي». فكرة النص التشعُّبي تسبق الشبكة العنكبوتية. وكان المهندس الأمريكي فانيفار بوش هو مَن كتب أول وصف لنظام تنظيم المعرفة عن طريق المستندات المترابطة، وظهر في عام ١٩٤٥ في صحيفة «أتلانتيك». أما شبكة الإنترنت العالمية — أو الشبكة العنكبوتية كما تُعرف الآن — فطوَّرها عالِم الكمبيوتر البريطاني تيم بيرنرز-لي في ثمانينيات القرن العشرين. كان بيرنرز-لي يعمل لدى المنظَّمة الأوروبية للأبحاث النووية (سيرن) خارج جينيف بسويسرا، وأراد أن ينشئ منظومةً تساعد العلماء على مشاركة المستندات والمعلومات. وتمكَّن العلماء من القيام بذلك من خلال إتاحة المستندات عبر الإنترنت، وكذلك إضافة روابط من مستنداتهم إلى مستندات أخرى كانت متاحة عبر الإنترنت. نمت الشبكة العنكبوتية واستمرت في النمو بالأساس بفضل الأشخاصِ الذين يضيفون صفحات ويب جديدة. فيكتب مؤلفو الشبكة العنكبوتية محتوى صفحات الويب ويربطونها بالصفحات الحالية ذات الصلة بمحتوى الصفحات التي يكتبونها.

تخيَّل أنك مؤلِّف لمقالٍ على الإنترنت يقدِّم نظرةً عامة حول تأثيرات تغيُّر المناخ في بلدك. قد ترغب، في معرض تقديمك لموضوع المقال، في جعل القراء يطَّلعون على صفحة ويب تعتقد أنها مصدرٌ موثوق فيه عن الموضوع نفسه؛ ومِن ثَم تضيف رابطًا إلى صفحة الويب تلك. وبهذه الطريقة، تساعد القراء عن طريق إتاحة الفرصة لهم للتعمُّق أكثر في الموضوع، وفي الوقت نفسه تضيف ثقلًا إلى محتواك الخاص لأنك تثبِت عباراتك بعبارات أخرى من صفحة ويب تثق بها.

هناك العديد مثلك ممن يكتبون مقالاتهم عبر الإنترنت عن تأثيرات تغيُّر المناخ في بلدانهم أو أقاليمهم. وقد يرغب كل واحد منهم أيضًا في إضافة روابط إلى مقالاتٍ يعتقدون أنها مصدرٌ موثوق فيه للموضوع الذي يكتبون فيه. ستنبثق الارتباطات التشعُّبية من هذه المقالات المكتوبة عبر الإنترنت للإشارة إلى مصادر معلومات ذات صلة.

إن السبب في ظهور صفحات وكالة ناسا في صدارة نتائج البحث عن تغيُّر المناخ هو أن كثيرًا من المؤلفين، يكتب كلٌّ منهم مقاله الخاص، قد قرروا إضافة ارتباط تشعبي لصفحة الويب الخاصة بناسا عن موضوع تغيُّر المناخ. لقد حدَّد المؤلفون اختياراتهم كلٌّ بمفرده، ولكن ربما اختار العديد منهم الصفحة نفسها، مثل صفحة ناسا. لذا من المنطقي اعتبار هذه الصفحة التي تتحدَّث عن تغيُّر المناخ مهمةً مقارنةً بصفحات الويب الأخرى.

يعمل النظام بأكمله كنوع من النظام الديمقراطي. فيربط مؤلفو صفحات الويب صفحاتهم بصفحات أخرى. وكلما زادت الروابط المشيرة إلى صفحة الويب تلك، اعتبرها المؤلفون مهمةً بما يكفي لإضافة رابطها على صفحاتهم الخاصة؛ ومِن ثَم تزداد أهميتها بوجه عام.

يعمل النظام بأكمله كنوع من الديمقراطية. فيربط مؤلفو صفحات الويب صفحاتهم بصفحات أخرى. وكلما زادت الروابط المشيرة إلى صفحة الويب تلك ... تزداد أهميتها بوجه عام.

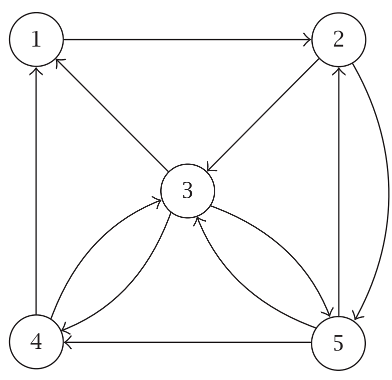
لكنَّ ثمَّة فرقًا مفاهيميًّا عن الديمقراطية التي نمارسها عادةً. فليست كل المقالات التي تُكتب متساوية. فبعضها يظهر على مواقع ويب مرموقة أكثر من غيره. فمقال على مدونة يقرؤه حَفنة من الناس يحمل ثقلًا أقل من مقال على موقع جهة من جهات النشر الإلكتروني يجتذب مئات الآلاف من القراء. وهذا يدل على أنه ينبغي ألَّا نتخذ عدد الروابط التي تشير إلى صفحة ويب بعينها مقياسًا لأهميتها. فالجهة التي تشير إلى صفحة الويب مهمة أيضًا، وليس مجرد عدد المُشيرين إليها. من المنطقي أن نتوقَّع أن يكون لرابط من صفحة ويب مرموقة ثِقلٌ أكبر من رابط من موقع مغمور. وعلى الرغم من أنه لا ينبغي الحكم على كتابٍ من غلافه، فإن المصادقة من مؤلِّف بارز أهم من مراجعة جيدة من مُشاهِد غير معروف. وكل رابط من صفحة إلى أخرى بمثابة مصادقة من الصفحة الأولى إلى الثانية، ويعتمد ثِقل المصادقة على وضع المصدق. في الوقت نفسه، إذا كانت صفحة تضيف روابط للعديد من الصفحات الأخرى، فينبغي أن تقسم مصادقتها — كما هو الحال — بين الصفحات التي تتلقَّاها.

تشكِّل مجموعة الصفحات المرتبطة بارتباطات تشعُّبية مخططًا بيانيًّا كبيرًا يحتوي على مليارات الصفحات والكثير الكثير من الروابط بينها. تمثِّل كل صفحة ويب عقدة في المخطط البياني. ويمثل كل رابط من صفحة إلى أخرى الحافة الموجهة في هذا المخطط

البياني الضخم. الفكرة الأساسية في خوارزمية بيج رانك هي أنه باتباع الاستدلال المنطقي الذي أوضحناه من قبل، يمكننا استخدام بنية المخطط البياني للشبكة العنكبوتية كي يوضح لنا أهمية كل صفحة. بعبارة أدقَّ، يمكننا معرفة أهمية كل صفحة من خلال رقم. وهذا الرقم — الذي سنسميه ترتيب الصفحة — سيقيس أهمية صفحة الويب مقارنة بصفحات الويب الأخرى. وكلما زادت أهمية صفحة الويب، ارتفعت قيمة ترتيب الصفحة. تتَّبع خوارزمية بيج رانك التشعُّب الناتج عن تلك الفكرة على نطاقٍ ضخم؛ أي على التمثيل البياني الذي يمثِّل الشبكة العنكبوتية بكاملها.

المبادئ الأساسية

عندما نكون على صفحة ويب، تشير الروابط في تلك الصفحة إلى صفحاتٍ أخرى ذات صلةٍ بمحتوى الصفحة التي نتصفحها في الوقت الحالي. ووجود ذلك الرابط في حد ذاته يدل على أهمية صفحة الويب الموجودة في نهاية الرابط، وإلا لما أضاف مؤلِّف صفحة الويب رابطها من البداية. انظر مثال المخطَّط البياني التالي الذي يمثِّل مجموعة صغيرة من صفحات الويب يرتبط بعضها ببعض:



في مخطَّط بياني كهذا، نسمي الروابط التي تشير إلى صفحة ويب «الروابط الخلفية»؛ وبالتبعية، سنسمي الصفحات التي تشير إلى صفحة ويب أيضًا «الروابط الخلفية». وبذلك تكون الروابط الخلفية لصفحة الويب رقم ٣ هي الحواف التي تشير إليها — الحواف المتجهة إليها — وهي كذلك العُقَد التي تبرز منها وهي صفحات الويب

رقم ٢ و٤ و٥. في هذا الفصل، سنهتم بالمخططات البيانية التي تتكوَّن من صفحات الويب؛ ومِن ثَم سنستخدم المصطلحين «عقدة» و«صفحة» بالتبادل.

سننشئ خوارزمية لإيجاد درجة أهمية كل صفحة ويب بناءً على مبدأين أساسيين وهما:

(١) اعتماد أهمية صفحة الويب على ثِقل صفحات الويب التي تذكر الرابط إليها؛ أي على أهمية روابطها الخلفية.

(٢) تقسيم صفحة الويب لأهميتها بالتساوي بين صفحات الويب التي تذكر الرابط إليها.

لنفترض أننا نريد إيجاد درجة أهمية الصفحة رقم ٣. رأينا أن الروابط الخلفية لها هي الصفحات ٢ و٤ و٥. نأخذ كل صفحة منها تِباعًا، ونفترض أننا نعرف ثِقلها. الصفحة ٢ تقسم أهميتها على الصفحتين ٣ و٥، وعليه سنعطي نصف أهميتها للصفحة رقم ٣. الصفحة ٤ أيضًا تقسم أهميتها على الصفحتين ٣ و١، وعليه سنعطي نصف أهميتها للصفحة رقم ٣. وأخيرًا، الصفحة ٥ تقسم أهميتها على الصفحات ٢ و٣ و٤، وعليه سنعطي ثلث أهميتها للصفحة رقم ٣. توفيرًا للكتابة، لنعبِّر بالرموز $r\ (p_i)$، حيث (i) تعبِّر عن أهمية الصفحة، بينما يرمز الحرف (r) إلى ترتيب الصفحة. إذن، قيمة أهمية الصفحة ٣ سوف تساوي:

$$r\ (P_3) = \frac{r\ (P_2)}{2} + \frac{r\ (P_4)}{2} + \frac{r\ (P_5)}{3}$$

بوجه عام، إذا كنا نريد حساب أهمية صفحة ويب معينة ونعرف أهمية كل ارتباط خلفي، يكون من السهل إيجاد ما نبحث عنه، من خلال قسمة درجة أهمية كل صفحة ذات رابط خلفي على عدد صفحات الويب التي تذكر الرابط إليها، ونضيف ناتج القسمة إلى مساهمات الروابط الخلفية الأخرى الخاصة بتلك الصفحة.

يمكن اعتبار حساب أهمية صفحات الويب كأنه سباق تصويتي بينها. كل صفحة مشاركة في التصويت لها ثِقَل يمكن أن تستخدمه كتأييد لصفحات الويب التي تعتبرها مهمة. إذا كانت تعتبر صفحة ويب واحدة هي المهمة، فإنها تعطي صوتَها تلك الصفحةَ. أما إذا رأت أن هناك أكثرَ من صفحة مهمة، فإنها تقسم صوتها وتعطي جزءًا منه كلَّ

صفحة ويب من تلك الصفحات. لذلك، إذا أرادت صفحة ويب التصويت لثلاث صفحات ويب أخرى باعتبارها مهمة، فستعطي كل صفحة منها ثلثَ صوتها. إلى أي الصفحات ستُخصص صفحة الويب صوتها؟ إلى الصفحات في نهاية الارتباطات التشعُّبية؛ أي الصفحات التي تضيف الرابط إليها. وكيف تُشتق أهمية صفحة الويب؟ تُشتق من أهمية روابطها الخلفية.

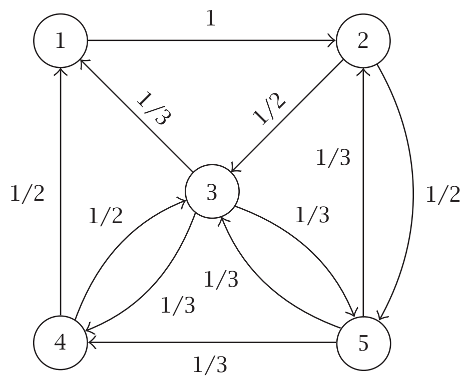
يضفي المبدآن بالفعل هالة من الديمقراطية على ترتيب صفحات الويب. فلا توجد سلطة منفردة تقرِّر الصفحات الأهم. فأهمية صفحة الويب تتحدَّد من أهميتها لدى صفحات الويب الأخرى التي تصوِّت عن طريق إضافة الروابط فيها. ولكن على النقيض من مبدأ شخص واحد يساوي صوتًا واحدًا المطبَّق في غالبية الانتخابات التي تجري في العالم الواقعي، ليست كل صفحات الويب تتساوى فيها الأصوات هنا. فأصوات صفحات الويب تعتمد على مدى أهمية الصفحة وتلك الأهمية تحدِّدها، مرة أخرى، صفحات الويب الأخرى.

قد يبدو هذا أشبه باحتيالٍ شرعي؛ لأنه في الواقع يخبرنا بأنه لا بد أن نجد أهمية الروابط الخلفية لصفحة ويب ما من أجل إيجاد أهميتها. إذا اتبعنا أسلوب الاستدلال نفسه لإيجاد أهمية كل رابط من الروابط الخلفية الخاصة بتلك الصفحة، فلا بد أن نوجد درجة أهمية الروابط الخلفية لذلك الرابط الخلفي. عندئذٍ تبدو العملية تتراجع أكثر وأكثر — من روابط خلفية إلى روابط خلفية — وفي النهاية نُترك دون معرفة كيفية حساب أهمية صفحة الويب من المكان الذي بدأنا من عنده. الأسوأ من ذلك أننا قد نجد أنفسنا ندور في دوائر. في المثال الذي بين أيدينا، كي نحسب أهمية الصفحة رقم ٣، نحتاج إلى معرفة درجة أهمية الصفحات ٢ و٤ و٥. ولحساب أهمية الصفحة ٢، نحتاج إلى أهمية الصفحة ١ (والصفحة ٥، ولكن لننحي هذا جانبًا الآن). ولحساب أهمية الصفحة ١، نحتاج إلى أهمية الصفحة ٤، ولإيجاد أهميتها، نحتاج إلى معرفة أهمية الصفحة ٣. لقد عُدنا إلى حيث بدأنا.

مثال

لمعرفة كيفية الخروج من هذه الإشكالية، لنفترض أننا نعطي كل الصفحات أهميةً متساوية قبل البدء في حساب أهمية صفحات الويب. بالاستعانة باستعارة التصويت المذكورة آنفًا، سنعطي كل صفحة ويب صوتًا واحدًا فقط. عند بدء التصويت، ستصوِّت كل صفحة من الصفحات بالطريقة التي ذكرناها، بتوزيع صوتها على الصفحات المرتبطة

بها. عندئذٍ ستتلقى كل صفحة الأصوات من كل روابطها الخلفية. سيبدو نقل الأصوات على النحو التالي:



ترسل الصفحة ١ صوتها إلى الصفحة ٢، وهي الصفحة الوحيدة المرتبطة بها. تقسم الصفحة ٢ صوتها إلى جزأين، وترسل النصف إلى الصفحة ٣، والنصف الآخر إلى الصفحة ٥. تقسم الصفحة ٣ صوتها إلى ثلاثة أجزاء، وترسل ثلثًا واحدًا إلى كلٍّ من الصفحات ١ و٤ و٥. وتصوِّت الصفحتان ٤ و٥ بالطريقة نفسها.

بمجرد انتهاء التصويت، ستحسب كل صفحة الإجمالي من مجموع الأصوات — أو كسور الأصوات — التي تلقتها من روابطها الخلفية. على سبيل المثال، بما أن الصفحة رقم ١ تلقت أصواتًا من الصفحتين ٣ و٤، فستكون درجة أهميتها ١ / ٢ + ١ / ٣ = ٥ / ٦ من الأصوات، وبما أن الصفحة ٣ تلقت أصواتًا من الصفحات ٢ و٤ و٥، فستحصل على ١ / ٢ + ١ / ٢ + ١ / ٣ = ٤ / ٣ من الأصوات. من ذلك نرى أن الصفحة رقم ١ قلَّلت حصتها من الأصوات مقارنةً بما بدأت به، بينما زادت الصفحة رقم ٣ من حصتها.

الآن، لنغير الإعداد قليلًا. بدلًا من إعطاء كل صفحة صوتًا واحدًا قبل بدء التصويت، سنعطي كل صفحة ١ / ٥ صوت بحيث يصبح مجموع الأصوات واحدًا. وبوجه عام، إذا كان لدينا العدد n من الصفحات، فسنعطي $1/n$ من الأصوات لكل صفحة منها. أما باقي العملية فسيظل كما هو بالضبط. إجمالي قيمة الأهمية لجميع صفحات الويب يساوي واحدًا، ومرة أخرى تُقسَّم الأهمية بالتساوي بين جميع صفحات الويب.

بعد انتهاء التصويت، ستكون درجة أهمية كل صفحة ويب قد تغيَّرت. فبدلًا من أن تساوي أهمية كل الصفحات ١ / ٥ = ٠٫٢، سنجد، بإجراء الحسابات، أنها تساوي

٠,١٧ و٠,٢٧ و٠,٢٧ و٠,١٣ و٠,١٧ لكل صفحة تباعًا. صفحتا الويب ٢ و٣ زادت قيمة أهميتهما، ولكن الصفحات ١ و٤ و٥ قلَّت قيمة أهميتها. حاصل جمع الأهمية الإجمالية لصفحات الويب يساوي واحدًا.

يمكننا الآن البدء في جولة تصويت جديدة باستخدام القواعد نفسها بحذافيرها. ستوزِّع الصفحات الأصوات التي جمعتها على الصفحات المرتبطة بها. وفي نهاية هذه الجولة الثانية، ستحصي كل صفحة أصواتها لتحديد موقفها من حيث درجات الأهمية المجمَّعة. بعد إجراء العمليات الحسابية، ستكون قيم الأهمية الجديدة كما يلي: ٠,١٦ و٠,٢٢ و٠,٢٦ و٠,١٤ و٠,٢٢.

سنعيد العملية مرة أخرى بحذافيرها. في الحقيقة، سنعيد عملية التصويت مرارًا وتكرارًا. وإذا فعلنا ذلك، فستتطور الأصوات — أو الأهمية المعينة لكل صفحة — كما هو مبيَّن في الجدول التالي، الذي يُظهر القيم الأولية والنتائج بعد كل جولة تصويت:

الجولة	الصفحة ١	الصفحة ٢	الصفحة ٣	الصفحة ٤	الصفحة ٥
البداية	٠,٢٠	٠,٢٠	٠,٢٠	٠,٢٠	٠,٢٠
١	٠,١٧	٠,٢٧	٠,٢٧	٠,١٣	٠,١٧
٢	٠,١٦	٠,٢٢	٠,٢٦	٠,١٤	٠,٢٢
٣	٠,١٦	٠,٢٣	٠,٢٦	٠,١٦	٠,٢٠
٤	٠,١٧	٠,٢٢	٠,٢٦	٠,١٥	٠,٢٠
٥	٠,١٦	٠,٢٣	٠,٢٥	٠,١٥	٠,٢٠
٦	٠,١٦	٠,٢٣	٠,٢٦	٠,١٥	٠,٢٠

إذا شرعنا في إجراء جولة تصويت سابعة، فسنكتشف أن الموقف لن يتغيَّر مقارنة بجولة التصويت السادسة. فستظل الأصوات كما هي دون تغيير؛ ومِن ثَم لن تتغيَّر أهمية صفحات الويب. وعندئذٍ نستخلص النتيجة النهائية. فيكون ترتيب صفحات الويب بأن الصفحة رقم ٣ هي الأهم، يتبعها الصفحة رقم ٢ ثم الصفحة رقم ٥ ثم الصفحة رقم ١ وأخيرًا الصفحة رقم ٤.

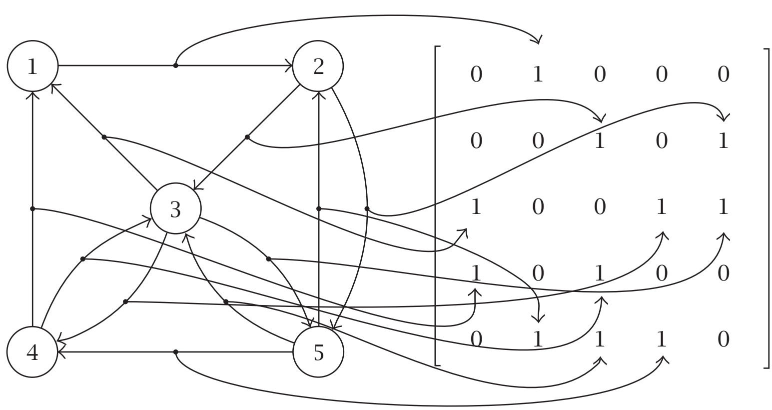
لنرجع خطوة إلى الوراء ونتأمَّل ما فعلناه. لقد بدأنا بمبدأين يحدِّدان قواعد لحساب أهمية صفحة الويب، شريطة أن نكون على علمٍ بأهمية كل رابط من الروابط الخلفية

للصفحة. قبل أن نبدأ، نَعدُّ كل صفحات الويب (n) بأهمية متساوية بحيث تساوي $1/n$. ثم نحسب ثقل كل صفحة ويب عن طريق جمع الأنصبة التي تحصل عليها من الروابط الخلفية للصفحة. تعطينا هذه العملية قيمًا جديدة لأهمية كل صفحة ويب مختلفة عن قيمة $1/n$ التي بدأنا من عندها. نعيد العملية بدءًا من تلك القيم. وينتج عن ذلك إيجاد مجموعة قيم أخرى. بعد تكرار هذه العملية عدة مرات، سنجد أن الموقف قد استقر؛ أي لن يتغيَّر مقياس الأهمية مع تكرار العملية. عندما نصل إلى تلك المرحلة، نسميها محطة التوقف ونعلن عن القيم التي توصَّلنا إليها.

السؤال هنا بالطبع هو ما إذا كان النهج الذي وصفناه لتونا يصلح بوجه عام وليس فقط في هذا المثال أم لا. إضافة إلى ذلك، هل يخرج لنا بنتائج منطقية؟

مصفوفة الارتباطات التشعُّبية وطريقة الأس

طريقة حساب أهمية صفحة ما بناءً على أهمية روابطها الخلفية لها صيغة ممتازة. نبدأ من المخطَّط البياني الذي يصف الروابط بين صفحات الويب. يمكننا التعبير عن المخطَّط البياني باستخدام «مصفوفة» أعداد نطلق عليها «مصفوفة التجاور» للمخطَّط البياني. بنية المصفوفة بسيطة وواضحة. ننشئ مصفوفة تحتوي على صفوف وأعمدة بعدد العُقَد في الرسم البياني. ثم نضع العدد واحد لكل تقاطع يتطابق مع رابطٍ ما والعدد صفر لكل التقاطعات الأخرى. فيما يلي مصفوفة التجاور لمثالنا:



يمكننا أيضًا التعبير عن أهمية صفحات الويب باستخدام صف أو «متجه» واحد:

$$[r\,(P_1) \quad r\,(P_2) \quad r\,(P_3) \quad r\,(P_4) \quad r\,(P_5)]$$

أمَا وقد دخلنا في الجوانب العملية لخوارزمية بيج رانك، سنبدأ في استخدام مصطلح ترتيب الصفحات للإشارة إلى أهمية صفحةٍ ما. سترى أن المصطلح سيكون له ما يبرِّره؛ إذ سنتمكَّن من اشتقاق ترتيب لكل الصفحات على الشبكة العنكبوتية بناءً على الأهمية. وبما أن الصف يحتوي على كل ترتيبات الصفحات، سنسميه «متجه ترتيب الصفحات» للمخطَّط البياني.

تُقسَّم أهمية صفحة الويب على الصفحات المرتبطة بها. وبما أن لدينا الآن مصفوفة التجاور، يمكننا تنفيذ تلك المهمة عن طريق الاتجاه إلى كل صف، وتقسيم كل قيمة في الصف على عدد القيم الموجودة في ذلك الصف. تلك الطريقة تعادل تقسيم صوت كل صفحة على عدد الروابط الخارجية التي تذكر تلك الصفحة. وإذا فعلنا ذلك، فسنحصل على المصفوفة التالية:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/3 & 1/3 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 1/3 & 1/3 & 1/3 & 0 \end{bmatrix}$$

نطلِق على تلك المصفوفة اسم «مصفوفة الارتباطات التشعُّبية».

إذا أمعنا النظر في مصفوفة الارتباطات التشعُّبية، فسنجد كل عمود يوضِّح كيفية اشتقاق أهمية الصفحة من الصفحات المرتبطة بها. لنأخذ العمود الأول المتعلِّق بأهمية الصفحة رقم ١. تستمد هذه الصفحة أهميتها من الصفحتين رقم ٣ و٤. الصفحة رقم ٣ تعطي ثلثَ أهميتها الصفحةَ رقم ١ لأنها ترتبط بثلاث صفحات، والصفحة رقم ٤ تعطي نصف أهميتها الصفحةَ ١ لأنها ترتبط بصفحتين. لا تتلقى الصفحة رقم ١ أهمية من

الصفحات الأخرى في التمثيل البياني لأن تلك الصفحات لا ترتبط بها. يمكننا التعبير عن ذلك بالمعادلة التالية:

$$r\,(P_1) \times 0 + r\,(P_2) \times 0 + \frac{r\,(P_3)}{3} + \frac{r\,(P_4)}{2} + r\,(P_5) \times 0$$
$$= \frac{r\,(P_3)}{3} + \frac{r\,(P_4)}{2}$$

ولكن هذا بالضبط هو تعريف $r\,(P_1)$، وهو ترتيب الصفحة رقم ١. نحن نحصل على ترتيب الصفحة بجمع حواصل ضرب العناصر لمتجه ترتيب الصفحات ذي العناصر المتطابقة في العمود الأول من متجه الارتباطات التشعُّبية.

لنرَ ما يحدث إذا أخذنا متجهَ ترتيب الصفحات وجمعنا حواصل ضرب عناصره مع العناصر المقابلة له في العمود الثاني من مصفوفة الارتباطات التشعُّبية:

$$r\,(P_1) \times 1 + r\,(P_2) \times 0 + r\,(P_3) \times 0 + r\,(P_4) \times 0 + \frac{r\,(P_5)}{3} = r\,(P_1) + \frac{r\,(P_5)}{3}$$

هذا بالضبط هو تعريف $r\,(P_2)$، وهو ترتيب الصفحة رقم ٢. بالمثل، جمع حواصل ضرب العناصر لمتجه ترتيب الصفحات في محتويات العمود الثالث من مصفوفة الارتباطات التشعُّبية سيعطينا $r\,(P_3)$، وهو ترتيب الصفحة رقم ٣:

$$r\,(P_1) \times 0 + \frac{r\,(P_2)}{2} + r\,(P_3) \times 0 + \frac{r\,(P_4)}{2} + \frac{r\,(P_5)}{3}$$
$$= \frac{r\,(P_2)}{2} + \frac{r\,(P_4)}{2} + \frac{r\,(P_5)}{3}$$

يمكنك التحقُّق من أن استخدام العمودين الرابع والخامس في مصفوفة الارتباطات التشعُّبية سيعطيك $r\,(P_4)$ و$r\,(P_5)$ على التوالي. وهذه العملية — جمع حواصل ضرب العناصر في متجه ترتيب الصفحات مع محتويات كل عمود في مصفوفة الارتباطات التشعُّبية — هي في الحقيقة حاصل ضرب متجه ترتيب الصفحات في مصفوفة الارتباطات التشعُّبية.

ما لم تكن على دراية بعمليات المصفوفة، قد يكون هذا أمرًا مربكًا؛ لأننا عادةً ما نتحدَّث عن حاصل ضرب عددين — وهي عملية الضرب المعروفة — وليس عن ناتج

بنيات مثل المتجهات والمصفوفات. يمكننا إسقاط العمليات الحسابية على البنيات الأخرى — وليس فقط الأرقام — ما دامت تناسبها. فحاصل ضرب المتجه في المصفوفة عبارة عن عملية حسابية. لا توجد ألغاز في المسألة: إنها ببساطة عملية نعرِّفها بأنها عملية حسابية خاصة تتضمَّن عناصر المتجه وعناصر المصفوفة.

لنفترض أننا نُعدُّ مخبوزات بيجل وكرواسون تُباع مقابل ٢٠٠ و١٥٠ دولارًا على التوالي. لدينا متجران؛ يبيع الأول ١٠ قطع بيجل و٢٠ قطعة كرواسون، ويبيع الثاني ١٥ قطعة بيجل و١٠ قطع كرواسون. كيف نوجد إجمالي المبيعات لكل متجر؟

لإيجاد إجمالي المبيعات من المتجر الأول، سنضرب سعر البيجل في عدد القطع المبيعة في ذلك المتجر، ونضرب سعر الكرواسون في عدد القطع المبيعة ثم نجمع الناتجين:

$$2.00 \times 10 + 1.50 \times 20 = 50$$

ونقوم بالعملية نفسها لإيجاد إجمالي مبيعات المتجر الثاني:

$$2.00 \times 15 + 1.50 \times 10 = 45$$

للتعبير عن هذه المعادلة بصورةٍ أكثرَ إيجازًا، نكتب أسعار البيجل والكرواسون في صورة متجه:

$$\begin{bmatrix} 2.00 & 1.50 \end{bmatrix}$$

نكتب أيضًا المبيعات اليومية في صورة مصفوفة. ستحتوي المصفوفة على عمودين، عمود لكل متجر، وصفين، أحدهما للبيجل والآخر للكرواسون:

$$\begin{bmatrix} 10 & 15 \\ 20 & 10 \end{bmatrix}$$

لإيجاد إجمالي المبيعات لكل متجر، نضرب عناصر المتجه في كل عمود من مصفوفة المبيعات ثم نجمع حواصل الضرب. تلك العملية تحدِّد حاصل ضرب المتجه في المصفوفة:

$$[2.00 \quad 1.50] \times \begin{bmatrix} 10 & 15 \\ 20 & 10 \end{bmatrix}$$

$$= [2.00 \times 10 + 1.50 \times 20 \quad 2.00 \times 15 + 1.50 \times 10] = [50 \quad 45]$$

حاصل ضرب المتجه في المصفوفة حالة خاصة لحاصل ضرب مصفوفتين. لنوسع نطاق المثال وبدلًا من أن يكون لدينا متجه يحتوي على أسعار البيجل والكرواسون، يصبح لدينا مصفوفة تضم الأسعار والأرباح لكل عملية مبيعات.

$$\begin{bmatrix} 2.00 & 1.50 \\ 0.20 & 0.10 \end{bmatrix}$$

لإيجاد إجمالي المبيعات لكل متجر وإجمالي الأرباح لكل متجر، سننشئ مصفوفة بحيث تكون المدخلات في الصف i والعمود j هي جمع حاصل ضرب الصف i من مصفوفة الأسعار والأرباح في الصف j من مصفوفة المبيعات. ويكون هذا هو تعريف حاصل ضرب المصفوفتين:

$$\begin{bmatrix} 2.00 & 1.50 \\ 0.10 & 0.20 \end{bmatrix} \times \begin{bmatrix} 10 & 15 \\ 20 & 10 \end{bmatrix}$$

$$= \begin{bmatrix} 2.00 \times 10 + 1.50 \times 20 & 2.00 \times 15 + 1.50 \times 10 \\ 0.10 \times 10 + 0.20 \times 20 & 0.10 \times 15 + 0.20 \times 10 \end{bmatrix} = \begin{bmatrix} 50 & 45 \\ 5 & 3.5 \end{bmatrix}$$

نعود إلى ترتيب الصفحات؛ في كل جولة، تعبِّر العملية الحسابية لمتجه ترتيب الصفحات في الحقيقة عن حاصل ضرب قيمة متجه ترتيب الصفحات في الجولة السابقة في مصفوفة الارتباطات التشعُّبية. ومع خوض الجولات، نحصل على تقديرات متعاقبة لترتيبات الصفحات؛ أي تقديرات تعاقبية لمتجه ترتيب الصفحات الذي يتكوَّن من تلك القيم. للحصول على تلك التقديرات المتعاقبة في متجه ترتيب الصفحات، لا نحتاج إلَّا إلى

ضرب المتجه في كل جولة في مصفوفة الارتباطات التشعُّبية، ومِن ثَم نحصل على المتجه للجولة التالية.

في الجولة الأولى، نبدأ بمتجه ترتيب الصفحات الذي تساوي محتوياته جميعًا $1/n$، حيث n هي عدد الصفحات. إذا عبَّرنا عن متجه ترتيب الصفحات الأول بالرمز π_1، وعن متجه ترتيب الصفحات في نهاية الدورة الأولى بالرمز π_2 وعن مصفوفة الارتباطات التشعُّبية بالرمز H، فسنحصل على الصيغة التالية:

$$\pi_2 = \pi_1 \times H$$

في كل جولة، نستخدم متجه ترتيب الصفحات لتلك الجولة لحساب متجه ترتيب الصفحات للجولة التالية. وفي جولة التصويت الثانية حيث حصلنا على تقديرات ترتيب الصفحة الثالثة — أي متجه ترتيب الصفحات الثالث — نجري العملية الحسابية التالية:

$$\pi_3 = \pi_2 \times H = (\pi_1 \times H) \times H = \pi_1 \times \ (H \times H) = \pi_1 \times H^2$$

في جولة التصويت الثالثة، نحصل على متجه ترتيب الصفحات الرابع:

$$\pi_4 = \pi_3 \times H = \left(\pi_1 \times H^2\right) \times H = \pi_1 \times \ \left(H^2 \times H\right) = \pi_1 \times H^3$$

وكما هو الحال في كل عملية تكرار، نضرب ناتج عملية التكرار السابقة في مصفوفة الارتباطات التشعُّبية، ونحصل في النهاية على سلسلة من حواصل ضرب التقديرات المتعاقبة لمتجه ترتيب الصفحات في مصفوفة الارتباطات التشعُّبية. وكما نرى، تلك العملية تكافئ ضرب متجه ترتيب الصفحات الأول في أسس متزايدة لمصفوفة الارتباطات التشعُّبية. يطلَق على عملية حساب التقديرات المتعاقبة هذه اسم «طريقة الأس». لذا نرى أن حساب ترتيبات الصفحات لمجموعة من صفحات الويب عبارة عن تطبيقٍ لطريقة الأس على متجه ترتيب الصفحات ومصفوفة الارتباطات التشعُّبية إلى أن يتوقَّف متجه ترتيب الصفحات الناتج عن التغيُّر، أو كما نقول، إلى أن «يتلاقى» مع قيمة ثابتة وهي المصفوفات النهائية لترتيبات الصفحات.

ها قد وصلنا إلى وصفٍ أدقَّ لوصف كيفية حساب ترتيبات الصفحات في التمثيل البياني للروابط بين الصفحات على الشبكة العنكبوتية:

(١) تكوين مصفوفة الارتباطات التشعُّبية للتمثيل البياني.

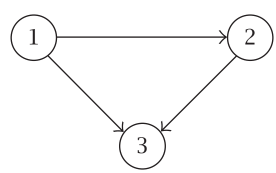
(٢) البدء من التقديرات الأولية لترتيب الصفحات، بحيث نعطي ترتيبًا بقيمة $1/n$ لكل صفحة، حيث n تعبِّر عن إجمالي عدد الصفحات.

(٣) تطبيق طريقة الأس، بضرب متجه ترتيب الصفحات في مصفوفة الارتباطات التشعُّبية إلى أن تثبت قيم متجه ترتيب الصفحات.

فضلًا عن الإيجاز الذي تتسم به هذه الصياغة، فإنها تتيح لنا نقل المسألة إلى ميدان علم الجبر الخطي، وهو فرع الرياضيات الذي يتعامل مع المصفوفات والعمليات التي تجري لها. توجد بنية ثابتة للنظرية يمكننا استخدامها لدراسة طريقة الأُس وكذلك تنفيذ عمليات المصفوفة، مثل الضرب الذي ذكرناه. كذلك ستساعد الصياغة المصفوفية للمسألة في معرفة إذا ما كانت طريقة الأُس ستتلاقى «دومًا» بحيث يمكننا التوصُّل دائمًا إلى حلٍّ لترتيبات الصفحات في التمثيل البياني أم لا.

نموذج العُقَد المتدلية والتصفُّح العشوائي

ننتقل الآن إلى مثالٍ لتمثيل بياني أبسط حيث يتكوَّن من ثلاث عُقَد فقط:



نريد إيجاد ترتيبات الصفحات لهذه العُقَد الثلاث. نتَّبع الخوارزمية نفسها. نضع القيمة الأولية لمتجه ترتيب الصفحات ١ / ٣، بحيث تتساوى ترتيبات الصفحات بين كل العُقَد. ثم نضرب متجه ترتيب الصفحات في مصفوفة الارتباطات التشعُّبية كما يلي:

$$\begin{bmatrix} 0 & 1/2 & 1/2 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

إذا بدأنا تكرارات طريقة الأس بحيث نضرب متجه ترتيب الصفحات في مصفوفة الارتباطات التشعُّبية لتحديث متجه ترتيب الصفحات، ثم أعدنا الكرَّة مرارًا، فسنجد أن جميع ترتيبات الصفحات بعد أربعة تَكرارات قد انخفضت إلى صفر:

الجولة	الصفحة ١	الصفحة ٢	الصفحة ٣
البداية	٠٫٣٣	٠٫٣٣	٠٫٣٣
١	٠٫٠٠	٠٫١٧	٠٫٥٠
٢	٠٫٠٠	٠٫٠٠	٠٫١٧
٣	٠٫٠٠	٠٫٠٠	٠٫٠٠

لا شك أن هذه مشكلة. فليس من المتوقَّع أن تكون أهمية جميع الصفحات هنا صفرًا. ففي النهاية، الصفحة رقم ٣ لها رابطان خلفيان، والصفحة رقم ٢ لها رابط خلفي واحد؛ ولذا توقَّعنا أن يظهر هذا بطريقةٍ ما في النتائج، فضلًا عن حقيقة أننا نريد أيضًا أن يساوي مجموع ترتيبات الصفحات واحدًا. أما هنا فلم نتوصَّل إلى وجود أهميةٍ لأي صفحة من الصفحات.

السبب في هذه المشكلة هو العقدة رقم ٣. على الرغم من أن هذه العقدة لها روابط خلفية، ومِن ثَم تكتسب أهمية، فلا يصدر منها أيُّ روابط. ولذا تستمد أهميةً بطريقةٍ ما من باقي التمثيل البياني ولكنها لا تعيد توزيعها على أي صفحة. إنها بمثابة عقدة أنانية أو ثقب أسود؛ فما يدخل إليها لا يخرج مرة أخرى. وبعد بضعة تكرارات، أصبحت بمثابة بالوعة دخلت فيها جميع قيم ترتيبات الصفحات واختفت.

يُطلق على مثل هذه العُقَد «العُقَد المتدلية»؛ لأنها تتدلَّى من النهايات (المغلقة) للتمثيل البياني. لا شيء يمنع وجود مثل تلك الصفحات على الشبكة العنكبوتية. وعلى الرغم من أن صفحات الويب تحتوي عادةً على روابطَ واردة وصادرة، فقد تظهر صفحة ليس لها روابط صادرة وتعيث فسادًا مع طريقة الأُس التي ذكرناها لتونا.

للتغلُّب على المشكلة، نستعين باستعارة. نتخيَّل أن لدينا شخصًا يتصفَّح الشبكة العنكبوتية ويقفز من صفحة إلى أخرى. للانتقال من صفحة إلى أخرى، عادةً ما يتَّبع الزائر رابطًا ما. ولكن بعد ذلك يصل المتصفح إلى عقدة متدلية؛ أي صفحة ليس لها روابط إلى أي صفحة أخرى. لا نريد للمتصفِّح أن يبقى حبيسًا في تلك الصفحة ومِن ثَم

نعطيه إمكانية القفز إلى أي صفحة أخرى، إلى أي مكانٍ على الشبكة العنكبوتية. الأمر أشبه بتصفح الشبكة العنكبوتية من صفحة إلى أخرى إلى أن نصل إلى طريق مسدود. وعندما يحدث ذلك، لا نستسلم ونتوقف. يمكننا دائمًا كتابة عنوان آخر في متصفح الويب وننتقل إلى أي صفحة ويب أخرى، حتى إن لم توجد لها روابط في الصفحة المتدلية. هذا ما نريد من المتصفِّح أن يفعله. عندما يضل وجهته، سيختار المتصفِّح صفحة — أي صفحة — على الشبكة العنكبوتية ويذهب إليها كي يستمر في التصفُّح. وهكذا يصبح المتصفِّح «متصفِّحًا عشوائيًّا» مزودًا بجهاز انتقال آنيٍّ يمكن أن يأخذه على الفور إلى أي مكان على شبكة الإنترنت.

لإسقاط تلك الاستعارة مرةً أخرى على ترتيب الصفحات، نفسِّر مصفوفة الارتباطات التشعُّبية بأنها تعطينا الاحتمالات بأن متصفِّحًا ما سيتَّبع رابطًا للانتقال إلى صفحةٍ بعينها. في مثال العُقَد الثلاث، يوضِّح الصف الأول من مصفوفة الارتباطات التشعُّبية أنه عندما يكون المتصفِّح في الصفحة رقم ١، ستتساوى احتمالات اختياره تصفُّح الصفحة رقم ٢ أو الصفحة رقم ٣. ويوضِّح الصف الثاني أنه عندما يكون المتصفِّح في الصفحة رقم ٢، فسيختار دومًا زيارة الصفحة رقم ٣. لنعُدْ إلى المثال الأول برهةً، إذا استقر المتصفِّح في الصفحة رقم ٥، فمن المحتمل أن ينتقل إلى الصفحة رقم ٢ أو ٣ أو ٤ لتكون احتمالية الزيارة ١ / ٣ لكلٍّ من تلك النتائج.

تعلن عقدة متدلية عن نفسها عندما يكون هناك صفٌّ مليء بالأصفار. عندئذٍ، لا توجد احتمالية أن ينتقل المتصفِّح إلى أي صفحة. وهنا يبرز تأثير المتصفِّح العشوائي. وكما ذكرنا، سيقفز ذلك المتصفح إلى أي صفحة في التمثيل البياني. وهذا يعني أننا في الواقع نغيِّر مصفوفة الارتباطات التشعُّبية بحيث لا تتبقى بها صفوف تحتوي على أصفار. وبما أننا نريد من المتصفِّح أن يقفز إلى أي صفحة ويب باحتمالية متساوية، سنكتب في الصف $1/n$ أو ١ / ٣ كما في المثال الذي معنا، بدلًا من الأصفار. وبذلك ستصبح المصفوفة بالشكل التالي:

$$\begin{bmatrix} 0 & 1/2 & 1/2 \\ 0 & 0 & 1 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

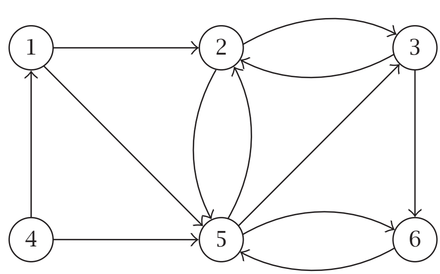
الآن، يمكن للمتصفِّح الذي يستقر في الصفحة رقم ٣ الانتقال إلى أي صفحة في التمثيل البياني باحتمالية متساوية. بل يمكن أن يبقى المتصفِّح في الصفحة نفسها مؤقتًا، ولكن هذا لا يهم؛ لأن الزائر يمكن أن يعيد المحاولة مرارًا، وفي وقتٍ ما سيتم اختيار صفحة أخرى عشوائيًّا. نطلِق على تلك المصفوفة مصفوفة الارتباطات التشعُّبية المعدَّلة، حيث يمكننا تغيير صفوف الأصفار إلى قيمٍ تساوي $1/n$، في المصفوفة S. إذا طبَّقنا طريقة الأُس باستخدام المصفوفة S، فستتطور ترتيبات الصفحات على النحو التالي:

الجولة	الصفحة ١	الصفحة ٢	الصفحة ٣
البداية	٠,٣٣	٠,٣٣	٠,٣٣
١	٠,١١	٠,٢٨	٠,٦١
٢	٠,٢٠	٠,٢٦	٠,٥٤
٣	٠,١٨	٠,٢٨	٠,٥٤
٤	٠,١٨	٠,٢٧	٠,٥٥
٥	٠,١٨	٠,٢٧	٠,٥٤

في هذه المرة، تتلاقى الخوارزمية عند قيمٍ غير صفرية؛ ومِن ثَم لا يحدث أيٌّ سحب للأهمية. كذلك تصبح النتائج منطقية. أعلى ترتيب صفحات حقَّقته الصفحة رقم ٣، برابطين خلفيين؛ ثم الصفحة رقم ٢ برابط خلفي واحد، ثم الصفحة رقم ١ التي لا يوجد لها أي روابط خلفية على الإطلاق.

مصفوفة جوجل

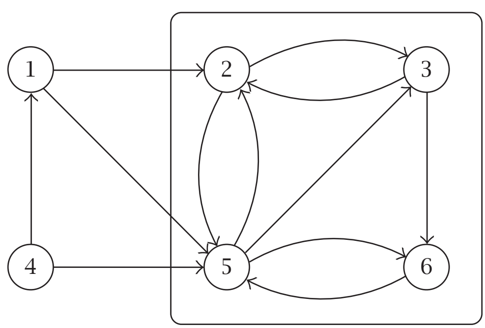
يبدو أننا حلَلنا الإشكالية، ولكن ثمَّة مشكلة مشابهة تطل برأسها في مواقفَ أعقد. لا يحتوي التمثيل البياني التالي على عُقَد متدلية:



إذا طبَّقنا الخوارزمية، فسنجد أن هناك عقدتين — الصفحة رقم ١ والصفحة رقم ٤ — تنتهيان بترتيب يساوي صفرًا:

الجولة	الصفحة ١	الصفحة ٢	الصفحة ٣	الصفحة ٤	الصفحة ٥	الصفحة ٦
البداية	٠,١٧	٠,١٧	٠,١٧	٠,١٧	٠,١٧	٠,١٧
١	٠,٠٨	٠,٢٢	٠,١٤	٠,٠٠	٠,٤٢	٠,١٤
٢	٠,٠٠	٠,٢٥	٠,٢٥	٠,٠٠	٠,٢٩	٠,٢١
٣	٠,٠٠	٠,٢٢	٠,٢٢	٠,٠٠	٠,٣٣	٠,٢٢

ما حدث أنه على الرغم من عدم وجود عقدة متدلية، توجد مجموعة عُقَد تعمل بمثابة بالوعة لباقي التمثيل البياني. إذا دقَّقت النظر في التمثيل البياني، فسترى أن العُقَد ٢ و٣ و٥ و٦، باعتبارها مجموعةً واحدة، لا تحتوي إلا على روابط واردة. يمكن الانتقال من العقدة ١ أو العقدة ٤ إلى هذه المجموعة، ولكن بمجرد الدخول إلى المجموعة، لا يمكننا سوى التنقُّل بداخلها. فلا يمكننا الخروج منها. وهكذا سيصبح المتصفح العشوائي أسيرًا، ليس داخل صفحة ويب واحدة هذه المرة، بل داخل مجموعة صفحات لا ترتبط إلا معًا.



مرة أخرى، ينبغي أن نساعد المتصفِّح العشوائي للخروج من هذا الفخ. ويتطلب الحل هذه المرة مزيدًا من التغييرات الشاملة في مصفوفة الارتباطات التشعُّبية. فمصفوفة الارتباطات التشعُّبية الأولية لا تتيح للمتصفِّح الانتقال من صفحة إلى أخرى إلا باستخدام

الروابط الموجودة في التمثيل البياني الأصلي. لذا عدَّلنا مصفوفة الارتباطات التشعُّبية للتعامل مع الصفوف التي تحتوي جميعًا على عناصر صفرية وتوصَّلنا إلى المصفوفة S التي أتاحت للمتصفِّح الابتعادَ عن العُقَد المتدلية. وقد أتاحت هذه الطريقة للمتصفِّح العشوائي أن يقفز إلى أي صفحة في التمثيل البياني عندما يقع في عقدة متدلية. الآن، سنغيِّر سلوك المتصفِّح العشوائي قليلًا عن طريق تعديل المصفوفة S.

في الوقت الحالي، عندما يستقر المتصفح عند عقدةٍ ما، تكون التحركات المحتملة هي المشار إليها في المصفوفة S. في المثال الأخير، المصفوفة S هي نفسها مصفوفة الارتباطات التشعُّبية نظرًا لعدم وجود صفوف صفرية:

$$\begin{bmatrix} 0 & 1/2 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 1/3 & 1/3 & 0 & 0 & 1/3 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

إذا استقر المتصفِّح العشوائي في الصفحة ٥، فستكون التحركات المحتملة إلى الصفحات ٢ أو ٣ أو ٦، باحتمالية تساوي ١ / ٣ لجميع الصفحات كما تشير المصفوفة S. سنجعل المتصفح العشوائي أكثر نشاطًا بحيث يتمكَّن من التحرُّك تبعًا للمصفوفة S، «ليس دائمًا»، ولكن باحتمالية قدرها a سنختارها؛ ومِن ثَم بالنسبة إلى الاحتمالية $(1 - a)$، سيقفز المتصفِّح العشوائي إلى أي صفحة في التمثيل البياني من دون التقيد بالمصفوفة S.

القدرة على التنقُّل من أي مكان إلى أي مكان في التمثيل البياني تعني أنه لا يمكن أن توجد قيم صفرية في المصفوفة على الإطلاق؛ لأن وجود مدخل صفري يشير إلى حركة لم تتم. ولتحقيق ما نريد، سنحتاج إلى «زيادة» المدخلات الصفرية في صف بقيمة ما و«تقليل» المدخلات غير الصفرية بحيث يكون مجموع الصف كاملًا هو العدد واحد دومًا.

يمكن حساب القيم النهائية في المصفوفة عن طريق الجبر الخطي بناءً على المصفوفة S والاحتمالية a. وتسمَّى المصفوفة الجديدة التي سنخرج بها «مصفوفة جوجل»، ونستخدم الرمز G للتعبير عنها. إذا تحدَّد سلوك المتصفِّح العشوائي وفقًا لمصفوفة جوجل، فسنخرج بالنتيجة التي نريدها: سيبدو المتصفِّح العشوائي كأنه يتبع المصفوفة S باحتمالية قيمتها a وينتقل بحرية بالاحتمالية $(1-a)$. في المثال الذي معنا، تصبح مصفوفة جوجل بالشكل التالي:

$$\begin{bmatrix} \frac{3}{120} & \frac{54}{120} & \frac{3}{120} & \frac{3}{120} & \frac{54}{120} & \frac{3}{120} \\ \frac{3}{120} & \frac{3}{120} & \frac{54}{120} & \frac{3}{120} & \frac{54}{120} & \frac{3}{120} \\ \frac{3}{120} & \frac{54}{120} & \frac{3}{120} & \frac{3}{120} & \frac{3}{120} & \frac{54}{120} \\ \frac{54}{120} & \frac{3}{120} & \frac{3}{120} & \frac{54}{120} & \frac{3}{120} & \frac{3}{120} \\ \frac{3}{120} & \frac{37}{120} & \frac{37}{120} & \frac{3}{120} & \frac{3}{120} & \frac{37}{120} \\ \frac{3}{120} & \frac{3}{120} & \frac{3}{120} & \frac{3}{120} & \frac{105}{120} & \frac{3}{120} \end{bmatrix}$$

قارن تلك المصفوفة بالمصفوفة S. لاحظ أنه في الصف الأول، كان لدينا مدخلان بقيمة ١ / ٢ وباقي المدخلات صفر. أما في مصفوفة جوجل، فتحوَّل المدخلان ١ / ٢ إلى ٥٤ / ١٢٠، وتحوَّلت باقي المدخلات من ٠ إلى ٣ / ١٢٠. وحدثت تغيُّرات مشابهة في الصفوف الأخرى. إذن، إذا استقر الزائر العشوائي في الصفحة رقم ١، تصبح التحرُّكات الممكنة إلى الصفحتين رقم ٢ و٥ باحتمالية ٥٤ / ١٢٠ لأي منهما، أو إلى أي صفحة أخرى باحتمالية ٣ / ١٢٠ لكل صفحة منها.

نستطيع الآن تقديم التعريف النهائي لخوارزمية بيج رانك:

(١) تكوين مصفوفة جوجل للتمثيل البياني.

(٢) البدء من التقديرات الأولية لترتيب الصفحات، بحيث نعطي ترتيب الصفحة بقيمة $1/n$ لكل صفحة، حيث n تعبِّر عن إجمالي عدد الصفحات.

(٣) تطبيق طريقة الأُس، بضرب متجه ترتيب الصفحات في مصفوفة جوجل إلى أن تتوقَّف قيم متجه ترتيب الصفحات عن التغيير.

ببساطة، وضعنا «مصفوفة جوجل» مكان «مصفوفة الارتباطات التشعُّبية» للخوارزمية الأولية. إذا تتبَّعنا هذه الخوارزمية في التمثيل البياني الذي يحتوي على مجموعة العُقَد البالوعية، فستكون النتيجة كما يلي:

الجولة	الصفحة ١	الصفحة ٢	الصفحة ٣	الصفحة ٤	الصفحة ٥	الصفحة ٦
البداية	٠,١٧	٠,١٧	٠,١٧	٠,١٧	٠,١٧	٠,١٧
١	٠,١٠	٠,١٤	٠,١٤	٠,١٠	٠,٣١	٠,٢١
٢	٠,٠٧	٠,١٥	٠,١٧	٠,٠٧	٠,٣١	٠,٢٣
٣	٠,٠٥	٠,١٤	٠,١٨	٠,٠٥	٠,٣٢	٠,٢٦
٤	٠,٠٥	٠,١٤	٠,١٧	٠,٠٥	٠,٣٣	٠,٢٧

إنها طريقة رائعة؛ إذ لم يَعُد لدينا ترتيبات صفحات صفرية.

تصلح طريقة الأُس مع مصفوفة جوجل على الدوام. فيوضح لنا الجبر الخطي أن المصفوفة ستتلاقى قيمها عند مجموعة نهائية من قيم ترتيب الصفحات، وسيكون مجموعها هو العدد واحد من دون معاناة مع العُقَد المتدلية أو وجود أجزاء في التمثيل البياني تستنزف ترتيبات الصفحات من بقية التمثيل البياني. بل إننا لا نحتاج إلى وضع قيم أولية لترتيبات الصفحات بحيث تصبح $1/n$ بالضبط عندما نبدأ. أي مجموعة من القيم الأولية ستفي بالغرض ما دام مجموعها يساوي واحدًا.

تطبيق خوارزمية بيج رانك عمليًّا

بعد إثبات أن لدينا طريقة لإيجاد ترتيب الصفحات في أي تمثيل بياني، يظل السؤال عمَّا إذا كانت النتائج في النهاية ستكون منطقية أم لا قائمًا.

يعتبر متجه ترتيب الصفحات — حسب التعريف الذي وضعناه له — متجهًا خاصًّا بالنسبة إلى مصفوفة جوجل. عندما تنهي طريقة الأس عملها، لا يتغير متجه ترتيب الصفحات بعد ذلك. ولذلك إذا ضربنا مصفوفة جوجل في متجه ترتيب الصفحات، فسنحصل ببساطة على متجه ترتيب الصفحات نفسه. في الجبر الخطي، يسمَّى هذا المتجه

«المتجه الذاتي الأول» لمصفوفة جوجل. من دون التعمُّق في الرياضيات، تدعم النظرية الأساسية التي تشكِّل هذا المتجه فكرة أنه يحظى بأهمية خاصة بالنسبة إلى المصفوفة.

بعيدًا عن الرياضيات، فإن الحكمَ الباتَّ فيما إذا كانت خوارزمية بيج رانك طريقةً جيدة في تعيين أهمية لصفحات الويب راجع إلى مدى استفادتنا — نحن البشرَ — من نتائجها. يعطينا محرك البحث جوجل نتائجَ جيدة، بمعنى أن النتائج تتطابق مع ما نراه — نحن مستخدمي محرِّك البحث — مُهمًّا. ولو كان متجه ترتيب الصفحات مجرَّد فضول رياضي لا علاقة له بأهمية صفحات الويب، لما اهتممنا به اليوم.

ثمَّة ميزة أخرى لخوارزمية بيج رانك وهي فاعلية تنفيذها. إن مصفوفة جوجل ضخمة؛ ونحن نريد صفًّا واحدًا وعمودًا واحدًا لكل صفحة على الشبكة العنكبوتية. ولكن مصفوفة جوجل، كما رأينا، مشتقة من المصفوفة S المشتقة بدورها من مصفوفة الارتباطات التشعُّبية. نحن لا نحتاج حقًّا إلى إنشاء مصفوفة جوجل نفسها وتخزينها؛ فبوسعنا إنشاؤها ديناميكيًّا باستخدام العمليات المصفوفية في مصفوفة الارتباطات التشعُّبية. وهذا أمر مريح وفي المتناول. وعلى النقيض من مصفوفة جوجل التي لا تحتوي على أي أصفار في أي مكان، تحتوي مصفوفة الارتباطات التشعُّبية على العديد والعديد من الأصفار. قد تتضمَّن الشبكة العنكبوتية مليارات الصفحات، ولكن كل صفحة لا ترتبط إلا بعدد محدود من صفحات الويب الأخرى. مصفوفة الارتباطات التشعُّبية هي ما نطلق عليه «مصفوفة متفرقة»؛ أي مصفوفة شبه مليئة بالأصفار ولا تحتوي إلا على عدد محدود من المدخلات غير الصفرية، وهي قيم أسية أقل من المدخلات الصفرية. ومِن ثَم يمكننا تخزين المصفوفة باستخدام أساليب ذكية بحيث لا تخزن سوى المواضع التي تظهر فيها المدخلات غير الصفرية بدلًا من الاحتياج إلى شريحة كبيرة من الذاكرة لملء معظمها بالأصفار وملء قدْر قليل منها بالمدخلات غير الصفرية. وبدلًا من تخزين مصفوفة الارتباطات التشعُّبية بأكملها، لا نحتاج إلا إلى تخزين إحداثيات المدخلات غير الصفرية التي لن تتطلب سوى جزء صغير جدًّا من مساحة التخزين. وهذا يمنحنا ميزة كبيرة في التطبيقات العملية لخوارزمية بيج رانك.

وأخيرًا، ثمَّة تنبيه مهم. على الرغم من أننا نعرف أن خوارزمية بيج رانك لعبت دورًا كبيرًا في نجاح شركة جوجل، فلا نعرف كيف تُستخدم خوارزمية بيج رانك في جوجل اليوم ولا حتى إن كانت مستخدمة أم لا. لقد ظل محرك البحث جوجل يتطور طيلة تلك السنين، والتغيُّرات التي تطرأ عليه لا تُعلن على الملأ. نعرف أن جوجل تستخدم عمليات

البحث التي أجريناها في الماضي كي تحسِّن النتائج التي تقدِّمها لاستفساراتنا. ويمكن أن تحسِّن النتائج بناءً على البلد الذي نعيش فيه. ويمكن أيضًا أن تضع في اعتبارها الاتجاهات الرائجة بوجه عام في عمليات البحث الأخرى التي يجريها الآخرون على مستوى العالم. كل ذلك جزء من الخلطة السرية التي تستخدمها جوجل لتحسين منتجها والحفاظ على مكانتها في عالَم محركات البحث أمام منافسيها. ولكن هذا لا ينتقص من فاعلية الخوارزمية في حل مسألة ترتيب صفحات الويب التي تتمثل في صورة عُقَد في التمثيل البياني.[1]

تُبرِز خوارزمية بيج رانك جانبًا آخرَ في الخوارزميات. إن نجاح الخوارزمية لا يُعلَّق فقط على روعتها وفاعليتها. بل يتعلَّق الأمر أيضًا بتخطيط الخوارزمية لحل مسألةٍ ما. وهذا نهج ابتكاري. فحل مسألة البحث في الشبكة العنكبوتية يستلزم التغلُّب على مشكلة الحجم الهائل للشبكة العنكبوتية. ولكن بمجرد تخيُّل الشبكة العنكبوتية في صورة تمثيل بياني، يتحوُّل حجمها إلى ميزة لا عقبة. ويعود السبب في ذلك تحديدًا إلى وجود كم هائل من الصفحات المرتبطة بعضها ببعض عن طريق الارتباطات التشعُّبية، حتى إنك قد تتوقَّع أن طريقة قائمة على بنية الارتباطات في التمثيل البياني سوف تنجح في النهاية. وإيجاد طريقة لوضع نموذج للمسألة هو الخطوة الأولى لإيجاد طريقة لحلها باستخدام خوارزمية.

الفصل السادس

التعلُّم العميق

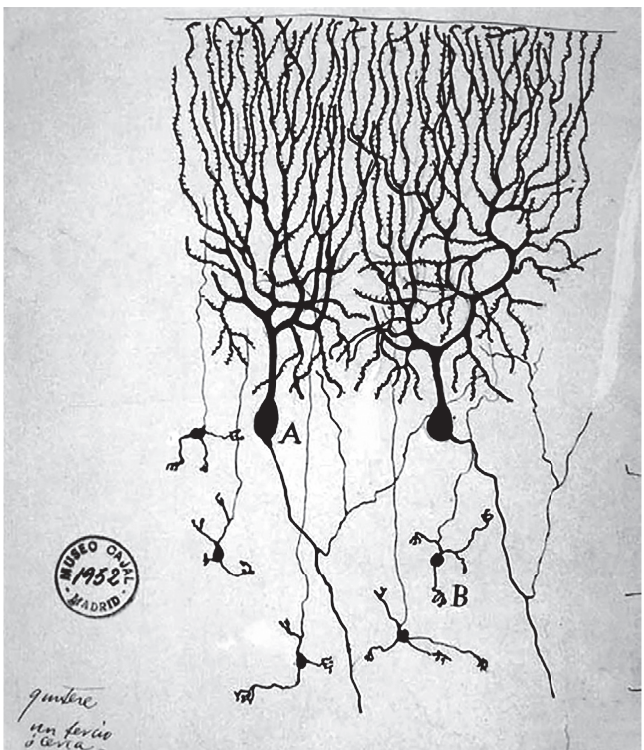
برزت أنظمة التعلُّم العميق فجأةً على الساحة في السنوات الأخيرة، وكثيرًا ما تصدَّرت العناوين في وسائل الإعلام الرئيسية. أصبحنا نرى أنظمة الكمبيوتر تنفِّذ مهامَّ فذةً استأثر بها البشر يومًا ما. الأعجب أن تلك الأنظمة كثيرًا ما تُقدَّم بوصفها تحمل أوجهَ تشابهٍ بينها وبين آلية عمل العقل البشري؛ وهو ما يشير بالطبع إلى فكرة أن أساس الذكاء الاصطناعي قد يكون قائمًا على محاكاة الذكاء البشري.

بغض النظر عن الدعاية المُبالغ فيها، فإن معظم العلماء الذين يعملون في مجال التعلُّم العميق لا يقولون إن أنظمة التعلُّم العميق تعمل مثل العقل البشري. فالهدف هو إظهار سلوك مفيد غالبًا ما نربطه بالذكاء. لكننا لسنا بصدد محاكاة الطبيعة؛ فبنية الدماغ البشري، في واقع الأمر، معقَّدة لدرجة يصعب تقليدها على جهاز كمبيوتر. ولكننا بالفعل نأخذ بعضًا من أوراق كتاب الطبيعة ونبسطها إلى حدٍّ كبير ونحاول أن نصمِّم أنظمةً تستطيع — في مجالات معينة — أن تنجز مهامَّ عادةً ما تنجزها الأنظمة البيولوجية التي تطوَّرت على مدى ملايين السنين. إضافة إلى ذلك، يمكن فهْم أنظمة التعلم العميق في إطار الخوارزميات التي تطبِّقها، وهذا ما يعنينا في هذا الكتاب. وهذا سيلقي بعض الضوء على المهام التي ينجزها بالتحديد وكيف ينجزها. ومن المفترض أن يساعدنا ذلك في إدراك أن الأفكار الرئيسة التي تقف وراء إنجازات تلك الأنظمة ليست أفكارًا معقَّدة. ولا ينبغي أن ينتقص هذا من الإنجازات التي تحققت في هذا المجال. فسنرى أن التعلم العميق يتطلب كمًّا هائلًا من الذكاء البشري كي يؤتي ثماره.

لفهم مضمون التعلم العميق، ينبغي أن ننطلق من بدايات بسيطة وسهلة. وعلى تلك البدايات سنبني صورةً أكثر تعقيدًا وتفصيلًا؛ إلى أن نتمكَّن، مع نهاية الفصل، من فهم ما تعنيه كلمة «عميق» في مصطلح التعلم العميق.

الخلايا العصبية، الحقيقية والاصطناعية

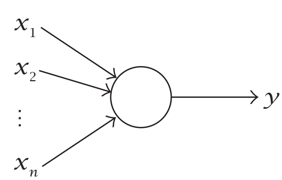
سنبدأ من حجر الزاوية الأساسي لأنظمة التعلم العميق، القادم من علم الأحياء. الدماغ جزء من الجهاز العصبي، والمكوِّنات الأساسية للجهاز العصبي عبارة عن خلايا تسمَّى الخلايا العصبية. تتميز الخلايا العصبية بشكل خاص؛ فهي تبدو مختلفة في الشكل عن البِنى الكروية التي عادةً ما نربطها بالخلايا. وفيما يلي واحدة من أولى صور الخلايا العصبية، رسمها العالم الإسباني سانتياجو رامون إي كاخال عام ١٨٩٩، أحد مؤسسي علم الأعصاب الحديث.[1]



البنيتان الظاهرتان في وسط الصورة عبارة عن خليتين عصبيتين في دماغ حمامة. كما ترى، تتكوَّن الخلية العصبية من جسم الخلية والخيوط التي تنبثق منها. تربط هذه

الخيوط الخلية العصبية بخلايا عصبية أخرى عبر «تشابكات عصبية» بطريقة تدمج الخلايا العصبية في شكل شبكة. الخلايا العصبية غير متماثلة. فيوجد العديد من الخيوط في جانب وخيط واحد فقط في الجانب الآخر لكل خلية عصبية. يمكننا القول إن الخيوط العديدة في الجانب الأول تمثِّل مدخلات الخلية العصبية، والخيط الطويل الخارج من الجانب الآخر يمثِّل مخرج الخلية العصبية. تستقبل الخلية العصبية المدخلات في شكل إشارات كهربية من تشابكاتها العصبية الواردة وقد ترسل إشارة إلى الخلايا العصبية الأخرى. وكلما زادت الإشارات الواردة، زاد احتمال أن ترسل هي إشارة. عندئذٍ نقول إن الخلية العصبية «محفزة» أو «نشطة».

الدماغ البشري عبارة عن شبكة ضخمة من الخلايا العصبية يبلغ عددها نحو مائة مليار خلية، وكل خلية منها متصلة في المتوسط بآلاف من الخلايا العصبية الأخرى. ليست لدينا الوسائل لبناء شبكة بهذا الحجم، ولكن يمكننا بناء أنظمة مستمَدة من نماذج مبسطة ومثالية من الخلايا العصبية. وفيما يلي نموذج لخلية عصبية اصطناعية:

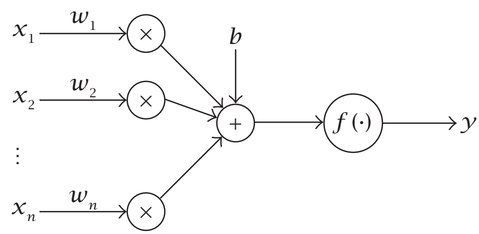


تلك نسخة تجريدية من خلية عصبية حيوية؛ مجرَّد بنية لها عدد من المدخلات ومخرج واحد. يعتمد مخرج الخلية العصبية الحيوية على مدخلاتها؛ وبالمثل، نريد تنشيط الخلية العصبية الاصطناعية بناءً على مدخلاتها. لسنا في مجال الكيمياء الحيوية للدماغ ولكننا في مجال الكمبيوتر؛ لذا نحتاج إلى نموذج حوسبي للخلية العصبية الاصطناعية. لنفترض أن الإشارات التي تستقبلها الخلايا العصبية وترسلها عبارة عن أرقام. عندئذٍ، تأخذ الخلية العصبية الاصطناعية كل المدخلات وتحسب قيمةً حسابيةً ما بناءً على تلك المدخلات، وتخرج لنا بنتيجةٍ ما على مخرجها. لا نحتاج إلى أي دارة خاصة لتنفيذ خلية عصبية اصطناعية. يمكنك تخيُّلها كبرنامج صغير على جهاز كمبيوتر يستقبل المدخلات ويحوِّلها إلى مخرجات، مثل أي برنامج آخر على الكمبيوتر. لسنا بحاجة إلى بناء شبكات عصبية اصطناعية بالمعنى الحرفي؛ بل يمكننا محاكاتها وهذا ما نقوم به بالفعل.

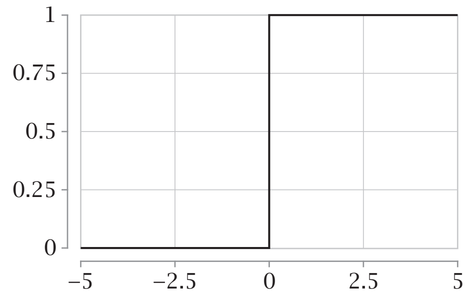
جزء من عملية التعلم في الشبكات العصبية البيولوجية يكمُن في تقوية التشابكات العصبية بين الخلايا العصبية أو إضعافها. فاكتساب قدرات معرفية جديدة واستيعاب المعلومات يؤديان إلى تقوية بعض التشابكات العصبية بين الخلايا العصبية، بينما يؤدي ذلك إلى إضعاف خلايا أخرى أو حتى إخمادها بالكامل. إضافة إلى ذلك، قد لا تؤدي التشابكات العصبية إلى تحفيز الخلية العصبية فحسب، بل إلى تثبيط نشاطها؛ وعندما تصل إشارة إلى ذلك التشابك، لا تحفز تلك الخلية العصبية. ولدى الأطفال الرُّضع تشابكات عصبية في أدمغتهم أكثر من الكبار. وتشذيب الشبكات العصبية داخل أدمغتنا جزء من النمو. ربما يمكن تشبيه دماغ الطفل الرضيع بكتلة من الرخام؛ كلما مرَّت علينا السنين، تُشذَّب تلك الكتلة جراء التجارب والأشياء التي نتعلمها، ويظهر لها شكل محدَّد.

في الخلية العصبية الاصطناعية، نُقدِّر مرونة التشابكات العصبية — ودورها المحفز أو المثبط — من خلال «أوزان» نخصصها للمُدخلات. في نموذج الخلية العصبية الاصطناعية، لدينا n من المدخلات — X_1، X_2، ...، X_n. نخصِّص لكل مُدخل منها وزنًا (W)، W_1، W_1، ...، W_n. يُضرب كل وزن في المدخل الموازي له. المُدخل الأخير الذي تستقبله الخلية العصبية هو مجموع حواصل ضرب $W_1X_1 + W_2X_2 + \cdots + W_nX_n$ نضيف لهذا المُدخل الموزون الانحياز b، الذي يمكنك اعتباره النزعة الطبيعية لدى الخلية العصبية للاستثارة أو التحفيز؛ وكلما زاد الانحياز، زاد ميلها إلى النشاط، بينما سيؤدي إضافة انحياز سلبي إلى الإدخال الموزون إلى تثبيط الخلية العصبية وعزوفها عن الاستثارة.

تعد قيم الأوزان والانحياز «معاملات» الخلية العصبية لأنها تؤثِّر في سلوكها. ولما كانت مخرجات الخلية العصبية الحيوية تعتمد على مدخلاتها، فإن مخرجات الخلية العصبية الاصطناعية تعتمد على المدخلات التي تستقبلها. وتتم تلك العملية عن طريق تغذية المدخلات في دالة تنشيط خاصة، التي تكون نتيجتها هي مخرجات الخلية العصبية. وهذا ما يحدث، على المستوى البياني، باستخدام الدالة $f(\cdot)$ كبديل لدالة التنشيط:



أبسط دالة تنشيط عبارة عن دالة درجية، تعطينا النتيجة ٠ أو ١. تُحفَّز الخلية العصبية وتعطي النتيجة ١ إذا كان المدخل إلى دالة التنشيط أكبر من صفر، أو تبقى النتيجة ثابتة على صفر إذا كان المدخل غير ذلك:

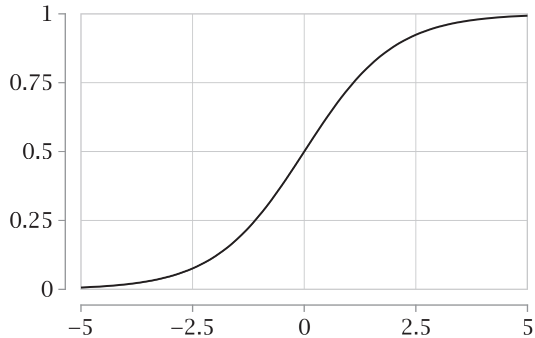


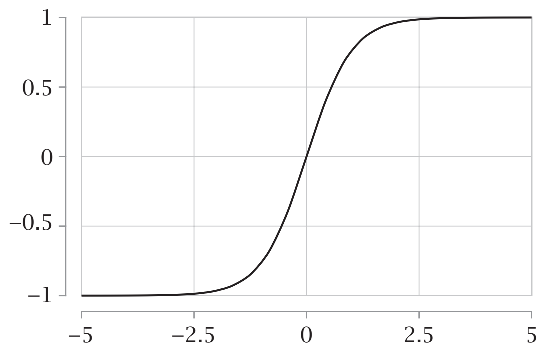
بدلًا من التفكير في الانحياز، من المفيد أن نفكِّر في حدٍّ. تعطي الخلية العصبية النتيجة ١ إذا كانت المدخلات الموزونة أعلى من الحد، وإذا كانت غير ذلك فستكون النتيجة صفرًا. في الحقيقة، إذا كتبنا سلوك الخلية العصبية في شكل صيغة، تكون الحالة الأولى هي $W_1X_1 + W_2X_2 + \cdots + W_nX_n + b > 0$ أو $W_1X_1 + W_2X_2 + \cdots + W_nX_n > -b$. وباستخدام $t = -b$، نحصل على النتيجة $W_1X_1 + W_2X_2 + \cdots + W_nX_n > t$، حيث t — معكوس الانحياز — هي الحد الذي تحتاج المدخلات الموزونة إلى تمريره إلى الخلية العصبية لتحفيزها.

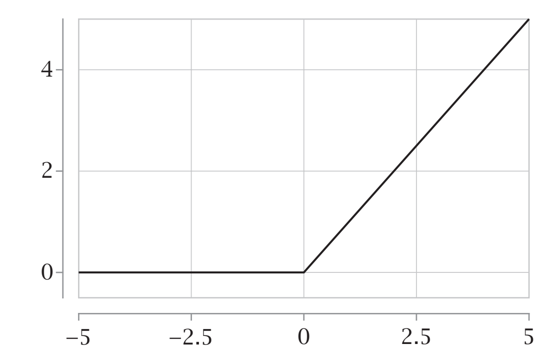
عمليًّا، نميل إلى استخدام دوال تنشيط أخرى ذات صلة بدلًا من الدالة الدرجية. ويمكن فيما يلي أن نرى ثلاث دوال شائعة.

تسمَّى الدالة الأولى «الدالة السينية»؛ لأنها تأخذ شكل حرف S.[2] وتتراوح مخرجاتها من صفر إلى ١. فالمدخلات الكبيرة الموجبة تعطي مخرجات قريبة من ١، والمدخلات الكبيرة السالبة تعطي مخرجات قريبة من صفر. وهذا يقارب خلية عصبية حيوية تُحفَّز مع المدخلات الكبيرة وتبقى ثابتة إذا كانت غير ذلك، كما أنه تقريب سلس إلى الدالة الدرجية. تسمَّى دالة التنشيط الثانية دالة الظل الزائدي.[3] إنها تشبه الدالة السينية، ولكنها تختلف في أن مخرجها يتراوح بين −١ و+١؛ إذ تؤدي المدخلات الكبيرة السالبة إلى نتيجة سالبة، محاكية بذلك إشارة تثبيط. الدالة الثالثة تسمَّى دالة المصحح؛ إذ تحول

كل المدخلات السالبة إلى صفر، وإلَّا فسيتناسب مخرجها طرديًّا مع مدخلاتها. يوضح الجدول التالي مخرجات دوال التنشيط الثلاث لمدخلات مختلفة.





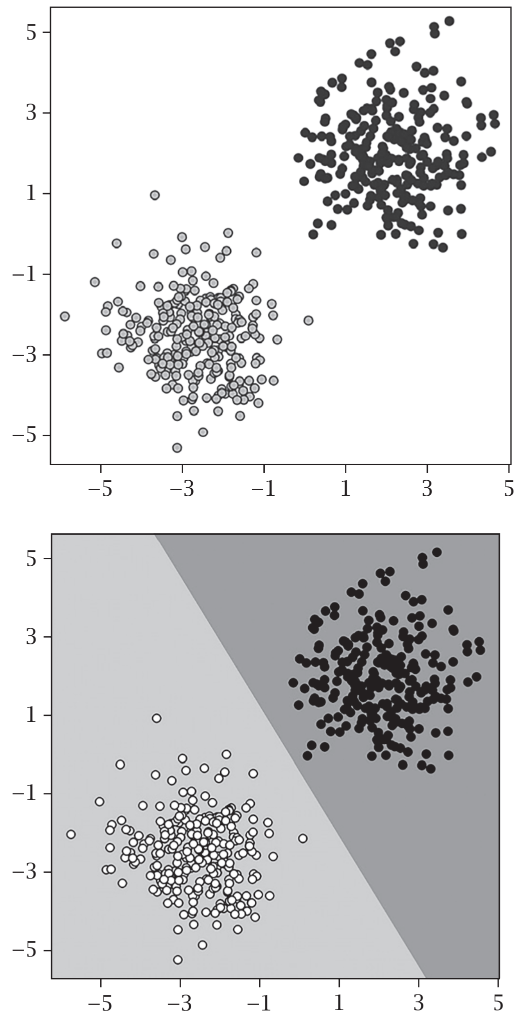


	٥−	١−	٠	١	٥
دالة سينية	٠,٠١	٠,٢٧	٠,٥	٠,٧٣	٠,٩٩
دالة الظل الزائدي	١−	٠,٧٦−	٠	٠,٧٦	١+
دالة المصحح	٠	٠	٠	١	٥

إذا كنت تتساءل عن سبب انتشار دوال التنشيط (في ظل وجود دوال غيرها)، فالسبب هو أنه ثبت عمليًّا أن بعض دوال التنشيط أنسب في بعض التطبيقات من غيرها. ولما كانت دالة التنشيط بالغة الأهمية بالنسبة إلى الخلية العصبية، فغالبًا ما تسمَّى الخلايا العصبية باسم دوال التنشيط الخاصة بها. فالخلية العصبية التي تستخدم الدالة الدرجية تسمَّى «بيرسيبترون».[4] إذن، لدينا الخلية العصبية السينية والخلية العصبية ذات الظل الزائدي. كذلك نطلق على الخلايا العصبية «وحدات»، بينما تسمَّى الخلية العصبية التي تستخدم دالة المصحح الوحدة الخطية المصححة (أو ريلو).

يمكن أن تتعلَّم الخلية العصبية الاصطناعية الفردية التمييز بين مجموعتين من العناصر. على سبيل المثال، لنأخذ البيانات في الصورة الأولى فيما يلي التي تصوِّر مجموعة من الملاحظات بعلامتين، x_1 على المحور الأفقي وx_2 على المحور الرأسي. نريد أن نبني نظامًا يستطيع التفريق بين مجموعتي النقاط. بناءً على أي عنصر، سيتمكَّن النظام من تحديد إذا ما كان العنصر يقع في إحدى المجموعتين أو الأخرى. في الواقع، سينشئ النظام «حدًّا للقرار»، كما في الشكل الثاني. بالنسبة إلى أي مجموعة مكوَّنة من (x_1، x_2)، سيخبرنا النظام إذا ما كان العنصر ينتمي إلى المجموعة ذات اللون الفاتح أم ذات اللون القاتم.

لن تتضمَّن الخلية العصبية أكثرَ من مدخلين. ستأخذ كل زوج (x_1، x_2) وتحسب مخرجًا. إذا كنَّا نستخدم دالة التنشيط السينية، فسيتراوح المخرج بين ٠ و١. سنأخذ القيم الأكبر من ٠,٥ وندرجها في إحدى المجموعتين ونأخذ القيم الباقية وندرجها في المجموعة الأخرى. بهذه الطريقة، سوف تعمل الخلية العصبية كمصنف، يصنف البيانات إلى فئتين مختلفتين. ولكن كيف تفعل ذلك؟ كيف يمكن للخلية العصبية أن تصل إلى مرحلة القدرة على تصنيف البيانات؟



عملية التعلم

في لحظة إنشاء الخلية العصبية، لا تتمكَّن الخلية من التعرُّف على أي نوع من البيانات؛ بل «تتعلَّم» التعرُّف على البيانات. ويتم هذا التعلُّم بطريق الأمثلة. العملية برُمَّتها تشبه تعليمَ الطالبِ درسًا جديدًا بإعطائه مجموعةً كبيرة من المسائل عن موضوع ما مصحوبة

بالحلول. نطلب من الطالب أن يذاكر كلَّ مسألة وحلها. إذا كان الطالب مجتهدًا، فإننا نتوقَّع أن يتعلم كيفية الانتقال من المسألة إلى الحل بعد الاطلاع على عددٍ من المسائل، بل يتمكَّن من حل مسائل جديدة ذات صلة بالمسائل التي درسها، ولكنه سيحلها هذه المرة من دون أن يلجأ إلى أي حلول.

عندما نقوم بذلك، فإننا «ندرِّب» الكمبيوتر على إيجاد الحلول؛ ويطلَق على مجموعة المسائل النموذجية المحلولة «مجموعة بيانات التدريب». ويُعَد هذا مثالًا على «التعلم الموجَّه»؛ لأن الحلول توجِّه جهاز الكمبيوتر — مثل المشرف — نحو البحث عن الإجابات الصحيحة. والتعلم الموجَّه هو الشكل الأكثر شيوعًا من أشكال «تعلُّم الآلة»، وهو الفرع الذي يتعامل بالكامل مع طرقٍ ندرِّب من خلالها أجهزة الكمبيوتر على إنجاز المهام. بعيدًا عن التعلم الموجَّه، يشتمل تعلُّم الآلة على «التعلم غير الموجَّه»، حيث نزوِّد جهاز الكمبيوتر بمجموعةٍ من بيانات التدريب ولكن دون إرفاق أي حلول معها. توجد تطبيقات مهمة للتعلم غير الموجَّه، ومنها على سبيل المثال تجميع الملاحظات في مجموعات مختلفة (لا يوجد حل مسبق يشير إلى ماهية المجموعة الصحيحة). ولكن بوجه عام، يتسم التعلم الموجَّه بتأثيرٍ أقوى من التعلم غير الموجَّه؛ لأننا نقدِّم مزيدًا من المعلومات في أثناء التدريب. لذا لن نتناول سوى التعلُّم الموجَّه في هذا المقام.

> في لحظة إنشاء الخلية العصبية، لا تتمكَّن الخلية من التعرُّف على أي نوع من البيانات؛ بل «تتعلم» التعرُّف على البيانات. ويتم هذا التعلُّم بطريق الأمثلة.

بعد التدريب، غالبًا ما يمر الطالب ببعض الاختبارات للوقوف على مدى إتقانه للمادة. بالمثل، في تعلُّم الآلة، نعطي جهاز الكمبيوتر بعد التدريب مجموعةَ بيانات أخرى لم يرَها من قبل ونطلب منه حل «مجموعة بيانات الاختبار» تلك. بعد ذلك نقيِّم أداء نظام تعلُّم الآلة بناءً على مدى إجادته لحل المسائل في مجموعة بيانات الاختبار.

في مهمة التصنيف، يسير التدريب في التعلُّم الموجَّه بإعطاء الشبكة العصبية عددًا كبيرًا من الملاحظات (المسائل) ومعها الفئات الخاصة بها (الحلول). ونتوقَّع أن تتعلم الخلية العصبية بصورة أو بأخرى كيفية الانتقال من ملاحظة ما إلى التصنيف الخاص بها. عندئذٍ إذا أعطيناها ملاحظة لم ترَها من قبل، يفترض أن تصنِّفها بقدرٍ معقول من النجاح.

يتحدَّد سلوك الخلية العصبية تجاه أي مدخل حسب أوزانه وانحيازه. عندما نبدأ، نضع الأوزان والانحياز بقيمٍ عشوائية، دون أن تعلم الخلية العصبية شيئًا عنها، مثل طالب جاهل بلا أي معلومات. ثم نعطي الخلية العصبية مدخلًا واحدًا في شكل مجموعة زوجية (x_1، x_2). ستُنتِج الخلية العصبية مُخرجًا ما. بما أن لدينا قيمَ أوزان وانحياز عشوائية، ستكون المخرجات أيضًا عشوائية. لكننا نعلم الإجابة الصحيحة التي ينبغي أن تصدر من الخلية العصبية بالنسبة إلى كل ملاحظة من الملاحظات في مجموعة بيانات التدريب. عندئذٍ، يمكننا حساب مدى بُعد مخرجات الخلية العصبية عن المخرجات المطلوبة. ويُطلَق على تلك النتيجة «الخسارة» وهي قياس درجة خطأ الخلية العصبية بالنسبة إلى مُدخل معيَّن.

على سبيل المثال، إذا نتج عن مدخلات الخلية العصبية مخرجًا قيمته ٠٫٢ في حين أن المُخرج المطلوب هو ١٫٠، يمكننا حساب الخسارة عن طريق طرح القيمتين إحداهما من الأخرى. وتجنبًا للاضطرار إلى التعامل مع العلامات، عادةً ما نعتبر الخسارة تربيعَ ناتجِ الطرح؛ وفي هذا المثال ستكون $(١٫٠ - ٠٫٢)^٢ = ٠٫٦٤$. فلو كان المُخرج المطلوب ٠٫٠، عندئذٍ ستصبح الخسارة $(٠٫٠ - ٠٫٢)^٢ = ٠٫٠٤$. أيًّا ما قد يكون الأمر، يمكننا الآن، بعد أن حسبنا الخسارة، تعديل الأوزان والانحياز لتقليلها.

نعود إلى الطالب، بعد كل محاولة فاشلة لحل تمرينٍ ما، نحثه على تقديم أداء أفضل. يكتشف الطالب أن عليه تغيير طريقته قليلًا ويحاول في المثال التالي. إذا فشل، نحثه مرة أخرى. ثم مرة أخرى. وبعد العديد من الأمثلة في مجموعة بيانات التدريب، سيبدأ في تصحيح الأمور أكثر وأكثر وسيتمكن من التعامل مع مجموعة بيانات الاختبار.

عندما يتعلم الطالب، يخبرنا علم الأعصاب أن الموصلات داخل الدماغ تتغير؛ فتقوى بعض التشابكات العصبية بين الخلايا العصبية وبعضها يضعف وبعضها الآخر يموت. لا يوجد مكافئ مباشر للخلية العصبية الاصطناعية، ولكن يحدث شيء مماثل. تذكَّر مرة أخرى أن سلوك الخلية العصبية يعتمد على مدخلاتها وأوزانها وانحيازها. ليس لدينا تحكُّم في المدخلات لأنها تأتي من البيئة. ولكن يمكننا تغيير قيم الأوزان والانحياز. وهذا ما يحدث في الحقيقة. فنحن نُحدِّث قيم الأوزان والانحياز بطريقةٍ تجعل الخلية العصبية تقلِّل من أخطائها.

الطريقة التي تستخدمها الخلية العصبية لتحقيق ذلك هي الاستفادة من ميزة طبيعة المهمة المطالبة بإنجازها. فنحن نريدها أن تأخذ كل ملاحظة وتحسب مُخرجًا

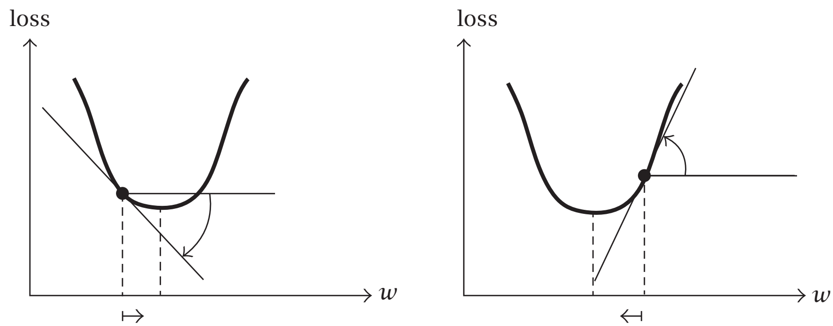
وفقًا للفئة، وتعدل أوزانها وانحيازها لتقليل الخسارة. إذن، تحاول الخلية العصبية حلَّ مسألة من مسائل «الحد الأدنى». بناءً على المدخلات والمخرجات التي تنتجها، تكون المسألة هي: «كيف لنا أن نعيد معايرة الأوزان والانحياز لتقليل الخسارة»؟

هذا يتطلب تغييرًا مفاهيميًّا للتركيز. حتى الآن، وصفنا الخلية العصبية بأنها شيء يأخذ بعض المدخلات ويحوِّلها إلى مُخرج. إذن فالخلية العصبية برُمَّتها، بهذا المعنى، هي عبارة عن دالة كبيرة تأخذ مدخلاتها وتطبِّق قيم الأوزان وتجمع حواصل الضرب وتجمع الانحياز وتمرِّر النتيجة عبر دالة التنشيط ثم تعطي المخرج النهائي. ولكن إذا فكَّرنا بطريقة أخرى، نجد أن المدخلات والمخرجات محدَّدة بالفعل (تلك هي مجموعة بيانات التدريب)، بينما ما يمكننا تغييره هو الأوزان والانحياز. ومِن ثَم يمكننا اعتبار الخلية العصبية بأنها دالة تتكوَّن متغيراتها من «الأوزان والانحياز»؛ لأننا لا نستطيع التأثير إلا في هذه القيم، ونريد أن نغيِّرها مع كل مُدخل للحد من الخسارة.

إذا أخذنا خلية عصبية بسيطة، كمثال توضيحي، لها وزن واحد وليس لها انحياز، فقد تكون العلاقة بين الخسارة والوزن كما في الجزء الأيسر من الشكل التوضيحي فيما يلي. يوضح المنحنى السميك الخسارة في صورة دالة وزن لُمدخل معيَّن. ينبغي أن تعدِّل الخلية العصبية وزنها بحيث تصل إلى أقل قيمة للدالة. بالنسبة إلى المدخل المعطى، تشتمل الخلية العصبية حاليًّا على خسارة في النقطة المشار إليها. للأسف، لا تعرف الخلية العصبية الوزن المثالي الذي من شأنه تقليل الخسارة بالنظر إلى أن الشيء الوحيد الذي تعرفه هو قيمة الدالة عند النقطة المشار إليها؛ إنها لا تملك منظورًا أفضل وأوسع مثلنا للشكل الموضَّح لنا. وقد لا تعدِّل الخلية العصبية الوزن إلا بقدر ضئيل — سواء بزيادة القيمة أو خفضها — بحيث تقترب من الحد الأدنى.

لمعرفة ما ينبغي فعله، سواء برفع قيمة الوزن أو خفضها، من الممكن أن توجِد الخلية العصبية خطَّ المماس عند النقطة الحالية. بعد ذلك يمكنها حساب ميل خط المماس؛ وهي الزاوية مع المحور الأفقي الموضحة أيضًا في الشكل. لاحظ أن الخلية العصبية يمكنها إجراء تلك العملية الحسابية دون أي إمكانيات خاصة سوى قدرتها على تنفيذ عمليات حسابية عند النقطة الموضعية. ميل خط المماس سالب؛ لأن الزاوية في اتجاه دوران عقارب الساعة. يوضح الميل «معدل التغيير في الدالة»؛ ومِن ثَم يشير الميل السالب إلى أنه مع زيادة الوزن، تقل الخسارة. عندئذٍ تكتشف الخلية العصبية أنه لتقليل الخسارة، ينبغي أن تتحرك إلى اليمين. وبما أن الميل سالب والتغيير المطلوب في الوزن موجب، تجد

الخلية العصبية أنه لا بد من تحريك الوزن في اتجاه موجب؛ أي اتجاه معاكس لما يشير إليه الميل.



ننتقل الآن إلى الشكل جهة اليمين. تقع الخلية العصبية هذه المرة جهة اليمين من الحد الأدنى للخسارة. تأخذ الخلية خطَّ المماس مرة أخرى وتحسب ميله. قيمة الزاوية موجبة، ومِن ثَم تكون قيمة الميل موجبة أيضًا. يشير الميل الموجب إلى أن زيادة الوزن تؤدي إلى زيادة الخسارة. عندئذٍ تعلم الخلية العصبية أنه يتعيَّن خفض قيمة الوزن لتقليل الخسارة. وبما أن الميل موجب والتغيير المطلوب في الوزن سالب، تجد الخلية العصبية مرة أخرى ضرورة التحرك في الاتجاه المعاكس للاتجاه الذي يشير إليه الميل.

إذن، القاعدة واحدة في الحالتين كلتيهما: تحسب الخلية العصبية الميل وتُحدِّث قيمة الوزن في الاتجاه المعاكس للميل. قد يكون كل هذا معروفًا من حساب التفاضل والتكامل. ميل الدالة عند نقطة ما هو «المشتقة» الخاصة بها. ولتقليل الخسارة، ينبغي تغيير الوزن بمقدار ضئيل عكس مشتق الخسارة.

لا تحتوي الخلية العصبية عادةً على وزن واحد، بل على عدة أوزان بالإضافة إلى الانحياز. ولمعرفة كيفية تعديل كل وزن على حدة وتعديل الانحياز، تبدأ الخلية العصبية مثلما ذكرنا في مثال الوزن الواحد. بمصطلحات رياضية، تحسب الخلية ما يسمَّى بـ «المشتقة الجزئية» للخسارة بالنسبة إلى كل وزن فردي والانحياز. بالنسبة إلى العدد n من الأوزان والانحياز، ستكون $n + 1$ من المشتقات الجزئية إجمالًا. يطلق على المتجه الذي يتضمَّن كل المشتقات الجزئية للدالة اسم «تدرُّج» الدالة. والتدرُّج هو مرادف الميل عندما يكون لدينا دالة متعددة المتغيرات؛ إنه يحدد الاتجاه الذي ينبغي أن نتبعه لزيادة

قيمة الدالة. ولتقليل قيمة الدالة، نسير في الاتجاه المعاكس. لذا لتقليل الخسارة، تحدِّث الخلية العصبية كل وزن والانحياز في الاتجاه المعاكس لما تشير إليه المشتقات الجزئية التي تشكِّل تدرُّجها.[5]

في الحقيقة لا تُجرى العمليات الحسابية برسم خطوط المماس وقياس الزوايا. توجد طرق فعَّالة لإيجاد المشتقات الجزئية والتدرُّج، ولكن لا داعي للخوض في التفاصيل. المهم هو أن لدينا طريقة محدَّدة لتعديل الأوزان والانحياز من أجل تحسين نتائج الخلية العصبية. وبذلك، يمكن وصف عملية التعلم بالخوارزمية التالية:

بالنسبة إلى كل مدخل وكل مُخرج مطلوب في مجموعة بيانات التدريب:

(١) يُحسب مُخرج الخلية العصبية والخسارة.
(٢) تُحدَّث قيم الأوزان والانحياز للخلية العصبية لتقليل الخسارة.

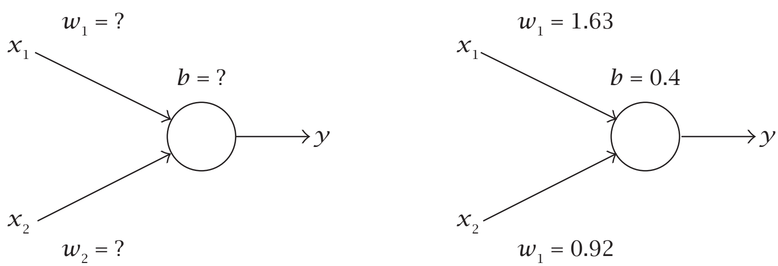
بمجرد الانتهاء من التدريب بالاطلاع على كل البيانات المدرجة في مجموعة بيانات التدريب، نقول إننا قد أكملنا «مرحلة». لكننا لا نتوقَّف عند هذا الحد عادةً. فالعملية تُعاد كاملة عدة مرات؛ إن الأمر أشبه بالطالب حينما يعيد مذاكرة المادة كلها من جديد بعد الاطلاع عليها كاملة. ومن المتوقَّع أن يحرز تحسنًا في المرة القادمة؛ لأنه هذه المرة لا يبدأ من الصفر — إذ لم يَعُد جاهلًا بالمادة — لأنه تعلَّم شيئًا بالفعل من المرحلة السابقة.

كلما كرَّرنا التدريب بإضافة مراحل إلى نظام التدريب، يزيد فهمنا لبيانات التدريب. لكن الإفراط في التدريب قد يكون شيئًا سيئًا. فالطالب الذي يدرُس مجموعةَ المسائل نفسها مرارًا ربما سيتعلم حلَّها من جذورها، دون أن يعرف كيفية حل أي مسائل أخرى لم يسبق أن تعرَّض لها. نرى ذلك يحدث عندما يفشل طالبٌ يبدو مستعدًّا جيدًا فشلًا ذريعًا في الامتحانات. في تعلُّم الآلة، عندما ندرِّب الكمبيوتر على مجموعةٍ معينة من بيانات التدريب، نقول إن التدريب «مناسب» للبيانات. أما التدريب المفرط فيؤدي إلى ما يسمَّى «فرط الاستعداد»: أي الإتيان بأداء ممتاز في مجموعة بيانات التدريب وأداء سيئ في مجموعة بيانات الاختبار.

يمكن إثبات أن الخلية العصبية، باتباع هذه الخوارزمية، يمكن أن تتعلَّم تصنيف أي بيانات «يمكن الفصل بينها خطيًّا». إذا كان للبيانات بُعدان (مثل المثال الذي معنا)، فهذا يعني أنه ينبغي الفصل بينها بخط مستقيم. أما إذا كان للبيانات سماتٌ أخرى، وليس

فقط (x_1، x_2)، فيُعمَّم المبدأ. بالنسبة إلى البيانات الثلاثية الأبعاد — أي ثلاثة مدخلات (x_1، x_2، x_3) — يمكن الفصل بينها خطيًّا إذا كان يمكن الفصل بينها بمستوًى بسيط في الفضاء الثلاثي الأبعاد. أما حال وجود مزيد من الأبعاد، فنطلق على المكافئ للخط والمستوى اسم «المستوى الفائق».

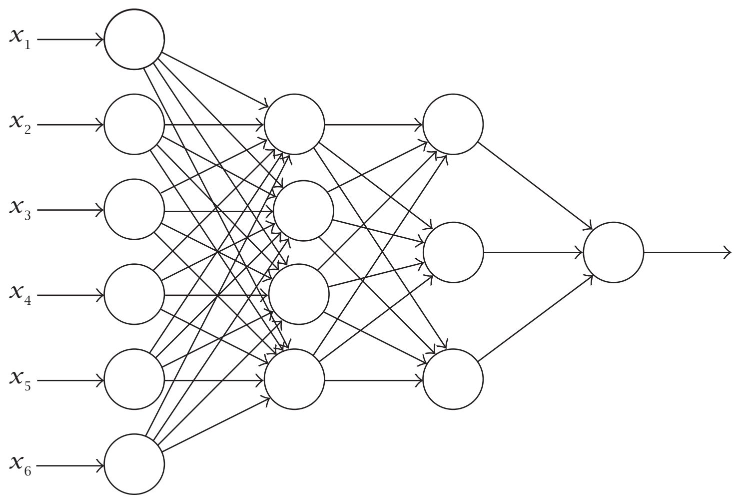
في نهاية التدريب، تكون الخلية العصبية قد تعلَّمت الفصل بين البيانات. وكلمة «تعلَّمت» هنا تعني أن الخلية العصبية وجدت الأوزان والانحياز المناسبين بالطريقة التي ذكرناها؛ أي بدأت بقيم عشوائية ثم حدَّثتها تدريجيًّا؛ ومِن ثَم قلَّت الخسارة. تذكَّر الشكل ذا مجموعتي النقاط حيث تعلَّمت الخلية العصبية الفصل باستخدام حد القرار. انتقلنا من الخلية العصبية جهة اليسار فيما يلي إلى الخلية العصبية جهة اليمين، حيث نرى القيم النهائية لمعاملاتها.



لا يحدث هذا دومًا. فالخلية العصبية المفردة التي تعمل بمفردها يمكنها تنفيذ مهامَّ محدَّدة فقط، مثل تصنيف البيانات القابلة للفصل خطيًّا. وللتعامل مع مهامَّ أعقدَ، نحتاج أن ننتقل من خلية عصبية اصطناعية وحيدة إلى شبكات الخلايا العصبية.

الانتقال من الخلايا العصبية إلى الشبكات العصبية

كما في الشبكات العصبية الحيوية، يمكننا بناء «شبكة عصبية اصطناعية» من خلايا عصبية مترابطة. يمكن ربط إشارات المدخلات لخلية عصبية بمخرجات خلايا عصبية أخرى، ويمكن أن ترتبط إشارة مخرجاتها بمدخلات خلايا عصبية أخرى. بهذه الطريقة يمكننا إنشاء شبكات عصبية كالتالية:

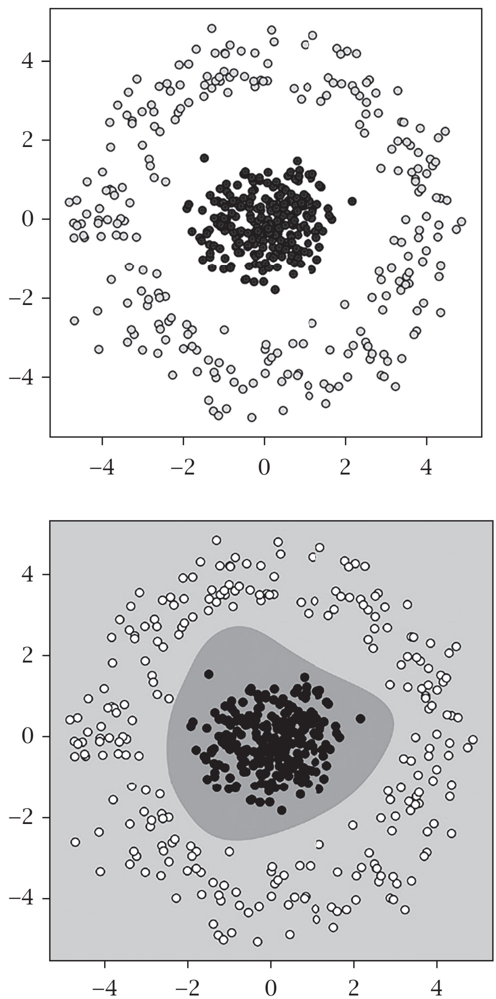


تُرتَّب الخلايا العصبية لهذه الشبكة العصبية الاصطناعية في شكل طبقات. وغالبًا ما يكون هذا هو المتَّبع في التطبيق العملي؛ حيث يتشكَّل العديد من الشبكات العصبية التي ننشئها من طبقات من الخلايا العصبية، كل طبقة مرصوصة بجوار الطبقة السابقة لها. كذلك جعلنا كل الخلايا العصبية في إحدى الطبقات متصلةً بكل الخلايا العصبية في الطبقة التالية، مع التحرُّك من اليسار إلى اليمين. مرة أخرى، تلك الطريقة شائعة على الرغم من عدم ضرورتها. عندما يكون لدينا طبقات متصلة بتلك الطريقة، نطلق عليها «طبقات كثيفة الاتصال».

على الرغم من أن الطبقة الأولى ليست متصلة بأي طبقة سابقة، فإن مخرجات الطبقة الأخيرة ليست مرتبطة بأي طبقة لاحقة بالمثل. فمخرجات الطبقة الأخيرة هي مخرجات الشبكة ككل؛ ومِن ثَم سوف توفِّر القيم التي نريد حسابها.

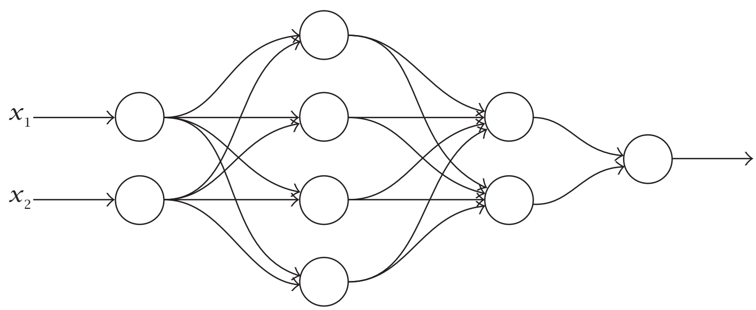
لِنرجعْ إلى مهمة التصنيف. تدور المسألة الآن حول الفصل بين مجموعتين من البيانات موضحتين في الشكل العلوي ممَّا يلي. تقع البيانات في دوائرَ متحدةِ المركز. واضح لأي لإنسان أن البيانات تنتمي لمجموعتين مختلفتين. وواضح أيضًا أنه لا يمكن فصلُ المجموعتين خطيًّا؛ فلا يمكن لخطٍّ مستقيم أن يفصل بين الفئتين. لذا نريد إنشاء شبكة عصبية تستطيع التمييز بين المجموعتين بحيث تخبرنا إلى أي مجموعة ستنتمي أي ملاحظة مستقبلية. هذا ما تراه في الشكل السفلي. بالنسبة إلى أي ملاحظة في الخلفية

ذات اللون الفاتح، ستدرك الشبكة العصبية أنها تنتمي إلى إحدى المجموعتين؛ وبالنسبة إلى أي ملاحظة في الخلفية ذات اللون القاتم، ستخبرنا أنها تنتمي إلى المجموعة الأخرى.



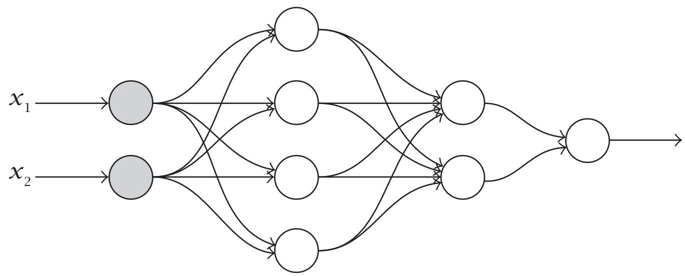
للوصول إلى النتائج التي نراها في الشكل السفلي، نبني شبكةَ طبقةٍ تلو الأخرى. ثم نضع خليتين عصبيتين في طبقة المدخلات، خلية لكل إحداثيٍّ للبيانات. نضيف طبقةً من

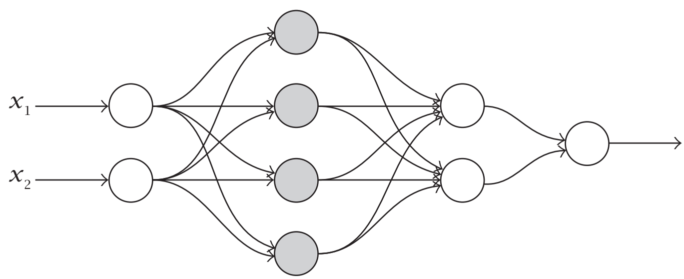
أربع خلايا عصبية كثيفة الاتصال بطبقة المدخلات. ونظرًا لأن هذه الطبقة ليست متصلة بالمدخلات أو المخرجات، فإنها «طبقة مخفية». نضيف طبقةً مخفية من خليتين عصبيتين مكثَّفة الاتصال بالطبقة المخفية الأولى. ننهي بناء الشبكة بطبقة مخرجات مكونة من خلية عصبية واحدة وكثيفة الاتصال بالطبقة المخفية الأخيرة. كل الخلايا العصبية تستخدم دالة التنشيط ذات الظل الزائدي. ستخرج طبقة المخرجات قيمة تتراوح بين −١ و+١، عارضة بذلك اعتقادها بأن البيانات تقع ضمن إحدى المجموعتين أو الأخرى. سنأخذ تلك القيمة ونحوِّلها إلى قرار ثنائي — إجابة تحتمل نعم أو لا — بناءً على إذا ما كانت تتجاوز ٠,٠ أم لا. وفيما يلي شكل الشبكة العصبية:

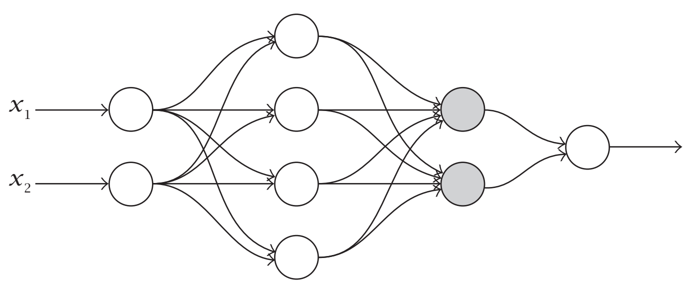


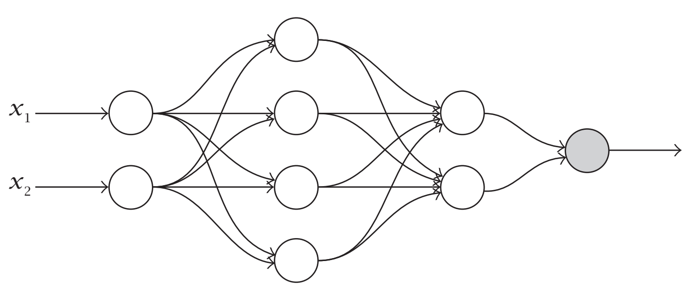
خوارزمية الانتشار العكسي

في البداية، لا تعرف الشبكة العصبية شيئًا، ولا يحدث تعديل؛ ومِن ثَم نبدأ بأوزان وانحيازات عشوائية. هذا ما يعنيه الجهل في عالم الشبكات العصبية. بعد ذلك نعطي الشبكة العصبية ملاحظة من البيانات التي معنا؛ أي مجموعة من الإحداثيات. سينتقل الإحداثيان x_1 وx_2 إلى طبقة المدخلات. تأخذ كلتا الخليتين العصبيتين القيمَ x_1 وx_2 وتمررانها باعتبارها مخرجاتهما إلى الطبقة المخفية الأولى. تحسب الخلايا العصبية الأربع في تلك الطبقة جميعًا مخرجاتها، وترسلها إلى الطبقة المخفية الثانية كل في دورها. ترسل الخلايا العصبية في تلك الطبقة مخرجاتها إلى الخلية العصبية في طبقة المخرجات، التي تنتج قيمة المخرجات النهائية للشبكة العصبية. مع تقدُّم العمليات الحسابية من طبقة إلى أخرى، تنشر الشبكة العصبية نتائج الخلايا العصبية للأمام من طبقة المدخلات إلى طبقة المخرجات:


x_1
x_2

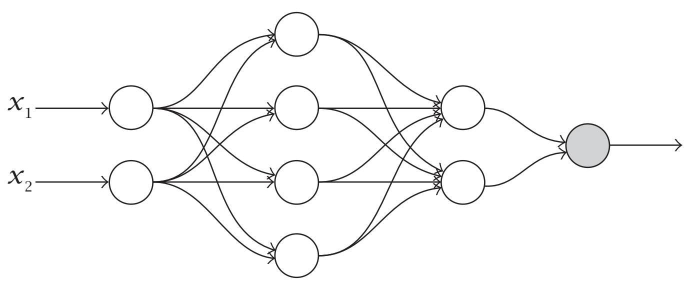

x_1
x_2

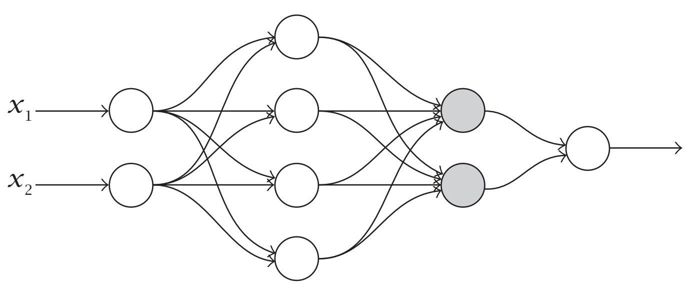

x_1
x_2


x_1
x_2

بمجرد الوصول إلى طبقة المخرجات، نحسب الخسارة مثلما فعلنا مع الخلية العصبية المفردة. وعندئذٍ لا نعدِّل قيم الأوزان والانحياز لخلية عصبية واحدة فحسب، بل للخلايا العصبية كافة في الشبكة من أجل تقليل الخسارة.

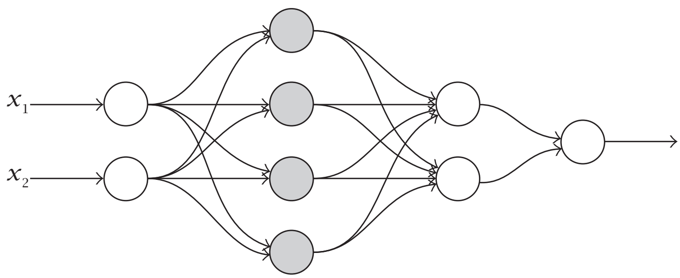
يتبيَّن أنه يمكن القيامُ بذلك بالتحرُّك في الاتجاه المعاكس بحيث نتَّجه من طبقة المخرجات إلى طبقة المدخلات. وبمجرَّد أن نعرِف قيمة الخسارة، يمكننا تحديث قيم الأوزان والانحياز للخلايا العصبية في طبقة المخرجات (لدينا هنا خلية عصبية واحدة، ولكن هذا المطلب ليس ضروريًّا على الدوام). بعد تحديث قيم الخلايا العصبية في طبقة المخرجات، يمكننا تحديث قيم الأوزان والانحياز للخلايا العصبية في الطبقة التي قبلها؛ أي الطبقة المخفية الأخيرة. بعد الانتهاء من ذلك، يمكننا تحديث قيم الأوزان والانحياز في الطبقة التي قبلها؛ أي الطبقة المخفية قبل الأخيرة. وهكذا إلى أن نصل إلى طبقة المدخلات:

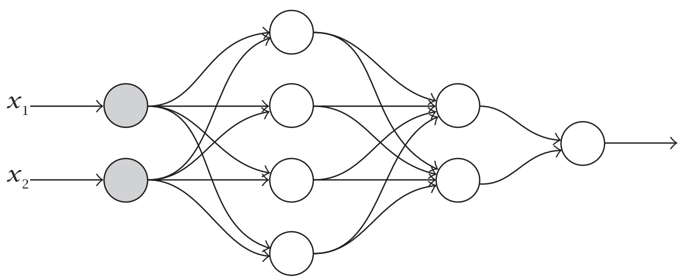




طريقة تحديث قيم الأوزان والانحياز للخلايا العصبية مماثلة لطريقة تحديثها في الخلية العصبية المفردة. مرة أخرى، تُحسب التحديثات بناءً على المشتقات الرياضية. يمكن تخيُّل شبكة عصبية كاملة كدالة كبيرة تتألَّف متغيراتها من قيم الأوزان والانحياز لكل الخلايا العصبية. بعد ذلك، يمكن حساب مشتقة كل وزن وانحياز فيما يتعلَّق

بالخسارة، واستخدام تلك المشتقة لتحديث الخلية العصبية. وبذلك نصل إلى صميم عملية التعلُّم في الشبكات العصبية ألا وهو: «خوارزمية الانتشار العكسي».[6]





بالنسبة إلى كل مُدخل والمُخرج المطلوب:

(١) تُحسب المخرجات والخسارة في الشبكة العصبية المتقدمة طبقةً تلو الأخرى بحيث نتقدَّم إلى الأمام من طبقة المدخلات إلى طبقة المخرجات.

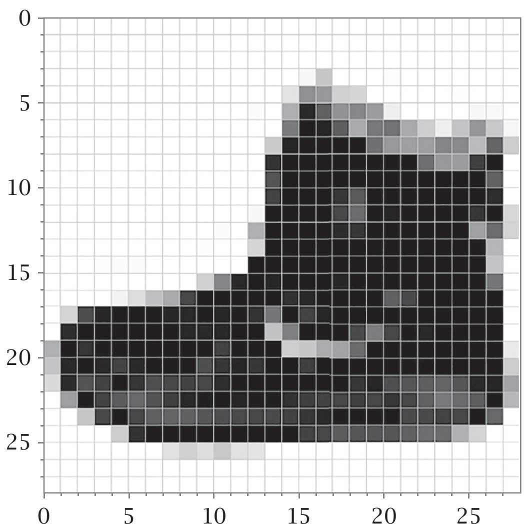
(٢) تُحدَّث قيم الأوزان والانحياز لتقليل الخسارة، بحيث نسير بترتيب عكسي من طبقة المخرجات إلى طبقة المدخلات.

باستخدام خوارزمية الانتشار العكسي، يمكننا بناء شبكات عصبية معقَّدة وتدريبها على تنفيذ مهامَّ مختلفة. لبنات أنظمة التعلم العميق بسيطة. إنها عبارة عن خلايا عصبية اصطناعية بقدرات حسابية محدودة؛ إذ تأخذ المدخلات وتضربها في الأوزان وتجمعها ثم تضيف قيمة الانحياز ثم تطبِّق إحدى دوال التنشيط على القيمة الناتجة. تستمد هذه الخلايا قوَّتها من الربط بين الكثير والكثير منها بطرق خاصة، حيث يمكن تدريب الشبكات الناتجة لتنفيذ المهمة التي نريد منها تنفيذها.

التعرُّف على الملابس

لجعل المناقشة أكثر واقعية، لنفترض أننا نريد بناء شبكة عصبية تتعرَّف على قطع ملابس معروضة في صور؛ ومِن ثَم سنسميها مهمة «التعرُّف على الملابس». لقد وُجد أن الشبكات العصبية تجيد تلك المهام على نحوٍ استثنائي.

ستكون كل صورة بحجمٍ صغير أبعادها ٢٨ × ٢٨. تتكوَّن مجموعة بيانات التدريب من ٦٠٠٠٠ صورة وتتكوَّن مجموعة بيانات الاختبار من ١٠٠٠٠ صورة؛ سنستخدم ٦٠٠٠٠ صورة لتدريب الشبكة العصبية و١٠٠٠٠ صورة أخرى لتقييم جودة التعلم. وفيما يلي مثال لصورةٍ أضفنا إليها مَحاور وشبكة كي تفيد في المناقشة فيما يلي:[7]



تنقسم الصورة إلى أجزاءٍ صغيرة مميزة لأننا نتعامل مع الصور رقميًّا بتلك الطريقة. نعتبر الصورة بأكملها مخطَّطًا مستطيلَ الشكل، ونقسِّمها إلى أجزاءٍ صغيرة بأبعاد ٢٨ × ٢٨ = ٧٨٤ ونعطي كل قطعة قيمة بعدد صحيح من ٠ إلى ٢٥٥، يوازي ظلًّا باللون الرمادي، حيث صفر يشير إلى اللون الأبيض بالكامل و٢٥٥ يشير إلى اللون الأسود بالكامل. الصورة السابقة هي في الواقع المصفوفة الواردة فيما يلي.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	13	73	0	0	1	4	0	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	3	0	36	136	127	62	54	0	0	0	1	3	4	0	0	3
0	0	0	0	0	0	0	0	0	0	0	0	6	0	102	204	176	134	144	123	23	0	0	0	0	12	10	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	155	236	207	178	107	156	161	109	64	23	77	130	72	15
0	0	0	0	0	0	0	0	0	0	0	1	0	69	207	223	218	216	216	163	127	121	122	146	141	88	172	66
0	0	0	0	0	0	0	0	0	1	1	1	0	200	232	232	233	229	223	223	215	213	164	127	123	196	229	0
0	0	0	0	0	0	0	0	0	0	0	0	0	183	225	216	223	228	235	227	224	222	224	221	223	245	173	0
0	0	0	0	0	0	0	0	0	0	0	0	0	193	228	218	213	198	180	212	210	211	213	223	220	243	202	0
0	0	0	0	0	0	0	0	0	1	3	0	12	219	220	212	218	192	169	227	208	218	224	212	226	197	209	52
0	0	0	0	0	0	0	0	0	0	6	0	99	244	222	220	218	203	198	221	215	213	222	220	245	119	167	56
0	0	0	0	0	0	0	0	0	4	0	0	55	236	228	230	228	240	232	213	218	223	234	217	217	209	92	0
0	0	1	4	6	7	2	0	0	0	0	0	237	226	217	223	222	219	222	221	216	223	229	215	218	255	77	0
0	3	0	0	0	0	0	0	0	62	145	204	228	207	213	221	218	208	211	218	224	223	219	215	224	244	159	0
0	0	0	0	18	44	82	107	189	228	220	222	217	226	200	205	211	230	224	234	176	188	250	248	233	238	215	0
0	57	187	208	224	221	224	208	204	214	208	209	200	159	245	193	206	223	255	255	221	234	221	211	220	232	246	0
3	202	228	224	221	211	211	214	205	205	205	220	240	80	150	255	229	221	188	154	191	210	204	209	222	228	225	0
98	233	198	210	222	229	229	234	249	220	194	215	217	241	65	73	106	117	168	219	221	215	217	223	223	224	229	29
75	204	212	204	193	205	211	225	216	185	197	206	198	213	240	195	227	245	239	223	218	212	209	222	220	221	230	67
48	203	183	194	213	197	185	190	194	192	202	214	219	221	220	236	225	216	199	206	186	181	177	172	181	205	206	115
0	122	219	193	179	171	183	196	204	210	213	207	211	210	200	196	194	191	195	191	198	192	176	156	167	177	210	92
0	0	74	189	212	191	175	172	175	181	185	188	189	188	193	198	204	209	210	210	211	188	188	194	192	216	170	0
2	0	0	0	66	200	222	237	239	242	246	243	244	221	220	193	191	179	182	182	181	176	166	168	99	58	0	0
0	0	0	0	0	0	0	40	61	44	72	41	35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

في الواقع، تتطلَّب الشبكات العصبية أن نقيس المدخلات على نطاق صغير من القيم — كأن يتراوح بين ٠ و١ — وإلا فقد لا تؤتي ثمارها؛ يمكن أن تفكِّر فيها باعتبارها تتضمَّن قيمَ مدخلات كبيرة تضلل الخلايا العصبية. هذا يعني أنه قبل استخدام هذه المصفوفة، كنَّا سنقسم كل خلية على ٢٥٥، ولكننا سنتجاهل ذلك في باقي المناقشة.

قد تنتمي قطع الملابس المختلفة إلى عشر فئات مختلفة يمكن أن تراها في الجدول التالي. أما بالنسبة إلى جهاز الكمبيوتر، فهذه الفئات ليست سوى أرقام مختلفة نطلق عليها «التسميات»:

التسمية	الفئة	التسمية	الفئة
٠	تي شيرت/ ملابس علوية	٥	صندل
١	سروال	٦	قميص
٢	بلوفر	٧	حذاء رياضي
٣	فستان	٨	حقيبة
٤	معطف	٩	حذاء كاحل

في الشكل التالي، نعرض عينة عشوائية لعشر قطع من كل نوع من الملابس. يوجد تنوُّع كبير في الصور كما ترى، وليست جميع الصور ذات جودة ممتازة في كل فئة ملابس بعينها. هذا يجعل المسألة مثيرة للاهتمام أكثر نوعًا ما. نريد إنشاء شبكة عصبية تأخذ صورًا كالموضَّحة في المثال مدخلاتٍ لها وتعطي مخرجاتٍ توضِّح نوع الصورة التي تعتقد أنها مُدخلها.

مرة أخرى، سنبني الشبكة العصبية في شكل طبقات. الطبقة الأولى — التي تتكوَّن من الخلايا العصبية المدخلة — ستتضمَّن ٧٨٤ خلية عصبية. ستأخذ كل خلية مُدخلًا واحدًا من جزء صغير واحد في الصورة، وببساطة ستخرج القيمة التي تحصل عليها في المُدخل الخاص بها. إذا كانت الصورة هي صورة حذاء الكاحل، فستحصل الخلية العصبية الأولى على القيمة في الجزء الصغير العلوي جهة اليسار — وهي صفر — في المُدخل، ومِن ثَم ستكون قيمة المُخرج صفرًا. ستحصل بقية الخلايا العصبية على قيم الأجزاء الصغيرة التي تسير مع اتجاه الصفوف؛ أي من الأعلى إلى الأسفل ومن اليسار إلى اليمين. الجزء الصغير ذو القيمة ٥٨ في الطرف الأيمن من كعب الحذاء (الصف الرابع

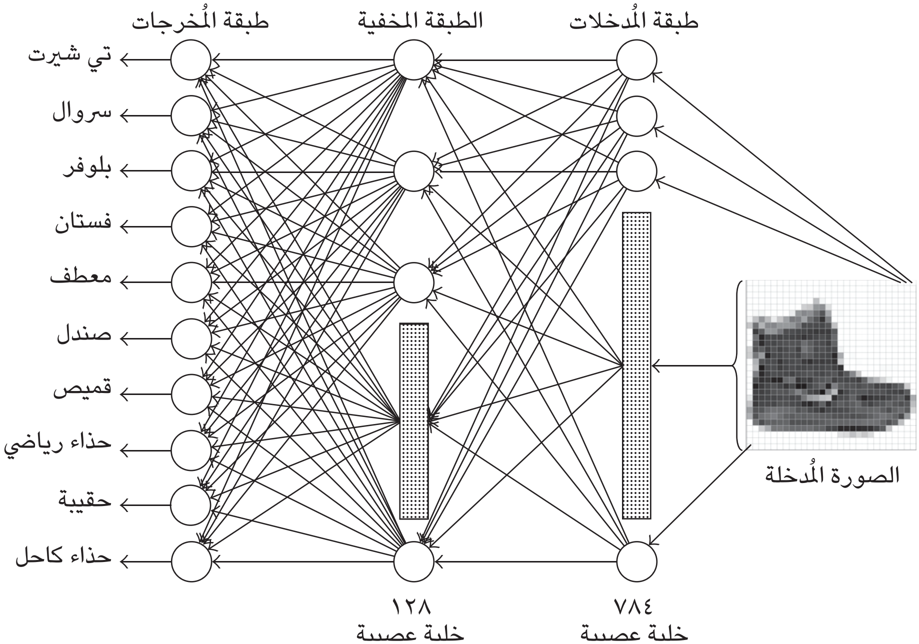
من الأسفل والعمود الثالث من اليمين) سيحصل على القيمة ٥٨ وينسخها في قيمة المُخرج الخاص به. بما أن الصفوف والأعمدة في الشبكة العصبية تُعد من الأعلى ومن اليسار، فإن هذه الخلية العصبية تقع في الصف الخامس والعشرين من الأعلى والعمود السادس والعشرين من اليسار، مما يجعلها خلية المُدخل رقم ٢٤ × ٢٨ + ٢٦ = ٦٩٨.

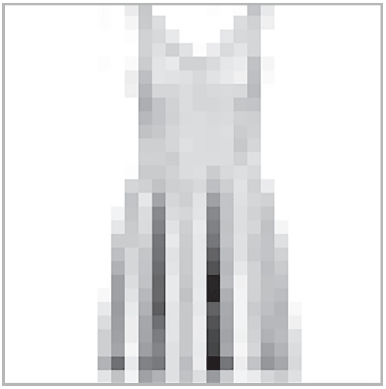
ستتصل الطبقة التالية اتصالًا مكثفًا بطبقة المُدخلات. ستحتوي على ١٢٨ خلية عصبية من خلايا الوحدة الخطية المصحَّحة. هذه الطبقة ليست متصلة مباشرةً بالصور المُدخلة (طبقة المُدخلات) ولن تتصل مباشرةً بالمُخرجات (سنضيف طبقةً أخرى للمُخرجات). لذا فهي طبقة مخفية؛ لأننا لا نستطيع ملاحظتها من خارج الشبكة العصبية. ونظرًا لأنها متصلة اتصالًا مكثفًا، فسيؤدي ذلك إلى عدد كبير من الروابط بين طبقة المدخلات والطبقة المخفية. ستتصل كل خلية عصبية في الطبقة المخفية بمخرجات كل الخلايا العصبية في طبقة المُدخلات. سيبلغ عدد روابط المُدخلات لكل خلية عصبية ٧٨٤، ومِن ثَم يكون الإجمالي ٧٨٤ × ١٢٨ = ١٠٠٣٥٢ رابطة.

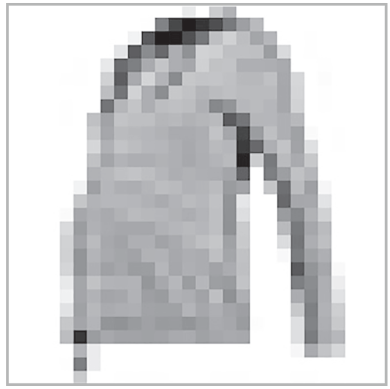
سنضيف طبقة أخرى أخيرة ستحتوي على خلايا المُخرجات العصبية التي ستحمل نتائج الشبكة العصبية. ستحتوي هذه الطبقة على ١٠ خلايا عصبية، بمعدل خلية لكل

فئة. ستتصل كل خلية عصبية خاصة بالمُخرجات بكل الخلايا العصبية في الطبقة المخفية، وبذلك يصبح إجمالي عدد الروابط ١٠ × ١٢٨ = ١٢٨٠ رابطة. المجموع الكلي لكل الروابط بين جميع الطبقات في الشبكة العصبية يساوي ١٠٠٣٥٢ + ١٢٨٠ = ١٠١٦٣٢. سيبدو شكل الشبكة العصبية الناتجة — بالشكل التخطيطي — كالشكل الموضَّح أدناه. ونظرًا لاستحالة تناسب كل العُقَد والحواف، يمكن أن تُرى مربَّعات منقَّطة تعبِّر عن مجموعة العُقَد في طبقة المُدخلات والطبقة المخفية؛ توجد ٧٨٠ عقدة في المربع الأول، و١٢٤ عقدة في المربع الثاني. أسقطنا كذلك الأسهم المتجهة إلى العُقَد الفردية داخل المربعات.

ستتكوَّن مُخرجات الشبكة العصبية من ١٠ مُخرجات، بمعدل واحد من كل خلية عصبية في الطبقة. ستعبِّر كل خلية عصبية مُخرجة عن فئة واحدة، وسيعبِّر إخراجها عن احتمالية أن تكون الصورة المُدخلة منتمية إلى تلك الفئة؛ ومِن ثَم سيكون مجموع احتمالات كل الخلايا العصبية العشرة ١، كما يجب أن يحدث عند التعامل مع الاحتمالات. يُعَد هذا مثالًا على دالة تنشيط أخرى تسمَّى «سوفت ماكس»، التي تأخذ المُدخلات في صورةِ متجهٍ لأعداد حقيقية وتحوِّلها إلى توزيع للاحتمالية. لننظر المثالين التاليين.







في المثال الأول على اليسار، نحصل على تلك النتيجة في مُخرجات الشبكة:

خلية المُخرجات	الفئة	الاحتمالية
١	تي شيرت/ملابس علوية	٠,٠٩
٢	سروال	٠,٠٣
٣	بلوفر	٠,٠٠
٤	فستان	٠,٨٣
٥	معطف	٠,٠٠
٦	صندل	٠,٠٠
٧	قميص	٠,٠٤
٨	حذاء رياضي	٠,٠٠
٩	حقيبة	٠,٠١
١٠	حذاء كاحل	٠,٠٠

هذا يعني أن الشبكة العصبية تخبِرنا بأنها متأكدة من أنها تتعامل مع فستانٍ معطيةً إياه نسبة احتمالية ٨٣ بالمائة، تاركةً نسبةً صغيرة من الاحتمالات جانبًا لأن تكون الصورة المُدخلة تي شيرت/ملابس علوية أو قميصًا أو سروالًا.

في المثال الثاني، على اليمين، تُخرِج لنا الشبكة الجدول التالي:

خلية المخرجات	الفئة	المخرجات
١	تي شيرت/ملابس علوية	٠,٠٠
٢	سروال	٠,٠٠
٣	بلوفر	٠,٣٣
٤	فستان	٠,٠٠
٥	معطف	٠,٢٤
٦	صندل	٠,٠٠
٧	قميص	٠,٤٣
٨	حذاء رياضي	٠,٠٠
٩	حقيبة	٠,٠٠
١٠	حذاء كاحل	٠,٠٠

الشبكة العصبية متأكدة بنسبة ٤٣ بالمائة من أنها تتعامل مع قميص، وهذا خطأ؛ فالصورة تعبِّر في الواقع عن بلوفر (في حالة أنك لم تتعرَّف عليها). ولكنها أعطت أفضل ثاني نسبة احتمال، وهي ٣٣ بالمائة، أن تكون الصورة لبلوفر.

ضربنا مثالًا توصَّلت فيه الشبكة إلى الإجابة الصحيحة ومثالًا آخر توصَّلت فيه الشبكة إلى الإجابة الخاطئة. وبوجه عام، إذا أعطينا الشبكة صورًا عديدة كي تتعرَّف عليها؛ أي الستين ألف صورة جميعها في مجموعة بيانات التدريب، فسنكتشف أنها تتمكَّن من التعرُّف على نحو ٨٦ بالمائة من ١٠٠٠٠ صورة في مجموعة بيانات الاختبار. وتلك نسبة ليست سيئة بالنظر إلى أن الشبكة العصبية لا تزال بسيطة على الرغم من كونها أكثر تعقيدًا بكثير من سابقتها. وانطلاقًا من هذه القاعدة، يمكننا إنشاء بِنيات لشبكاتٍ أعقدَ من شأنها أن تعطينا نتائج أفضل.

على الرغم من تزايد التعقيد، تتعلَّم الشبكة العصبية بالطريقة نفسها التي تتعرَّف بها الشبكات الأبسط على مجموعات البيانات والدوائر المتحدة المركز. نحصل على مخرجات عن كل عنصر من المُدخلات أثناء التدريب، ونقارن تلك المخرجات بالمخرجات المطلوبة

لحساب الخسارة. لم تَعُد المُخرجات قيمة واحدة الآن، بل عشر قيم، ولكن المبدأ واحد. عندما تتعرَّف الشبكة العصبية على قميصٍ بنسبة احتمال نحو ٨٣ بالمائة، يمكننا مقارنتها بالنسبة المثالية وهي التعرُّف عليه بنسبة احتمال ١٠٠ بالمائة. لذلك لدينا مجموعتان من قيم المُخرجات وهما: المجموعة التي نحصل عليها عن طريق الشبكة بنسب احتماليات متعددة تُعيَّن لمختلف أنواع الملابس، والمُخرجات التي نود الحصول عليها من الشبكة، وهي عبارة عن مجموعة من الاحتمالات جميعها تساوي صفرًا على عكس احتمالية واحدة، تطابق الإجابة الصحيحة التي تساوي واحدًا. في المثال الأخير، ستكون المُخرجات مقارنة بالنتيجة المستهدفة كالتالي:

خلية المُخرجات	الفئة	المخرجات	المستهدف
١	تي شيرت/ملابس علوية	٠,٠٠	٠,٠٠
٢	سروال	٠,٠٠	٠,٠٠
٣	بلوفر	٠,٣٣	١,٠٠
٤	فستان	٠,٠٠	٠,٠٠
٥	معطف	٠,٢٤	٠,٠٠
٦	صندل	٠,٠٠	٠,٠٠
٧	قميص	٠,٤٣	٠,٠٠
٨	حذاء رياضي	٠,٠٠	٠,٠٠
٩	حقيبة	٠,٠٠	٠,٠٠
١٠	حذاء كاحل	٠,٠٠	٠,٠٠

نأخذ العمودين الأخيرين ونحسب نسبة الخسارة مرةً أخرى؛ وهذه المرة فقط لا نحسب فرقًا تربيعيًّا بسيطًا نظرًا لعدم وجود قيمة مفردة. توجد مقاييس لحساب الفارق بين مجموعات القيم كتلك. وقد استخدمنا أحد تلك المقاييس في الشبكة العصبية، يسمَّى «الإنتروبيا المتقاطعة الفئوية» والذي يشير إلى مقدار الفرق بين توزيعين للاحتمالية. بعد حساب الخسارة، نُحدِّث الخلايا العصبية في طبقة المُخرجات. وبعد التحديث، نحدِّث الخلايا العصبية في الطبقة المخفية. باختصار، ننفِّذ خوارزمية الانتشار العكسي.

نخوض العملية نفسها برُمَّتها مع جميع الصور في مجموعة بيانات التدريب؛ أي مع مرحلة كاملة. وعند الانتهاء، نعيد كل ذلك في مرحلة أخرى. نكرِّر العملية مع محاولة تحقيق توازن: أي المرور بما يكفي من المراحل بحيث تكتسب الشبكة العصبية أكبرَ قدرٍ ممكن من مجموعة بيانات التدريب من دون المرور بعدد أكبر من اللازم من المراحل تكتسب فيها الشبكة العصبية قدرًا مفرطًا من مجموعة بيانات التدريب. في أثناء عملية التعلم، ستعدِّل الشبكة قيم أوزان وانحيازات خلاياها العصبية، وهي كثيرة. كلُّ ما تفعله طبقة المدخلات هو نسخ القيم إلى الطبقة المخفية، ومِن ثَم فلا حاجة إلى إجراء تعديلات في خلايا المُدخلات العصبية، ولكن يوجد ١٠٠٣٥٢ وزنًا في الطبقة المخفية و١٢٨٠ وزنًا في طبقة المُخرجات، و١٢٨ انحيازًا في الطبقة المخفية و١٠ انحيازات في طبقة المُخرجات، بمجموع ١٠١٧٧٠ معاملًا.

البدء في التعلم العميق

يمكن إثبات أنه على الرغم من أن الخلية العصبية لا يمكنها إنجاز الكثير بمفردها، فإن الشبكة العصبية يمكنها تنفيذ أي مهمة حسابية توصف لها بالخوارزميات وتشغَّل على جهاز كمبيوتر. ومِن ثَم لا توجد مهمة يستطيع الكمبيوتر فعْلها إلا وتستطيع الشبكة العصبية فعْلها. الفكرة العامة بالطبع هي أننا لا نحتاج إلى إخبار الشبكة العصبية بطريقة تنفيذ المهمة بالضبط. لا نحتاج سوى تغذيتها بأمثلة، وفي الوقت نفسه استخدام خوارزمية تتيح للشبكة العصبية تعلُّم طريقة تنفيذ المهمة. وقد رأينا أن خوارزمية الانتشار العكسي هي الخوارزمية الملائمة لذلك. على الرغم من أننا حصرنا أمثلتنا في التصنيف، يمكن تطبيق الشبكات العصبية على شتى أنواع المهام. فيمكنها التنبؤ بقيم كمية مستهدفة (مثل احتساب نقاط الجدارة الائتمانية) والترجمة بين اللغات وكذلك فهْم الكلام وإنشائه؛ وكذلك التغلُّب على أبطال البشر في لعبة «جو» مع إثارة حيرة الخبراء عن طريق توضيح استراتيجيات جديدة تمامًا لممارسة لعبةٍ مضى على عمرها قرون. بل استطاعت ممارسة لعبة «جو» مبتدئة فقط بمعرفة القواعد دون الوصول إلى مكتبة للمباريات التي جرت من قبل ثم الشروع في التعلم وكأن الشبكة العصبية كانت تلعب المباريات ضد نفسها.[8]

اليوم صارت التطبيقات الناجحة للشبكات العصبية كثيرة على الرغم من أن المبادئ ليست جديدة. فقد اختُرع البيرسيبترون في خمسينيات القرن العشرين، وعمر خوارزمية

الانتشار العكسي أكثر من ٣٠ عامًا. في تلك الفترة، ظهرت شبكات عصبية ثم بطَل استخدامها، مع تباين الحماس تجاه إمكانياتها بين ارتفاع وانخفاض. إن ما تغيَّر في السنوات القليلة الماضية، في الواقع، هو قدرتنا على بناء شبكات عصبية كبيرة حقًّا. وقد تحقَّق هذا بفضل التطورات في تصنيع رقاقات الكمبيوتر المتخصصة التي تمتاز بالكفاءة في تنفيذ العمليات الحسابية التي تقوم بها الخلايا العصبية. إذا تخيَّلت كل الخلايا العصبية في شبكة عصبية مرتَّبة داخل ذاكرة كمبيوتر، فإنه يمكن إجراء كل العمليات الحسابية اللازمة عن طريق عملياتٍ تعمل على مصفوفات ضخمة من الأعداد. تحسب الخلية العصبية مجموع حواصل الضرب الموزونة لمدخلاتها؛ إذا كنت تتذكَّر المناقشة حول خوارزمية بيج رانك في الفصل السابق، فإن مجموع حواصل الضرب هو جوهر ضرب المصفوفات.

وقد تبيَّن أن «وحدات معالجة الرسومات» مناسبة تمامًا لتلك المهمة. ووحدات معالجة الرسومات هي عبارة عن رقاقات كمبيوتر مصمَّمة خصيصى لإنشاء الصور داخل الكمبيوتر ومعالجتها؛ والمصطلح يُبنى على «وحدات المعالجة المركزية»، وهي الرقاقة التي تنفِّذ تعليمات برنامجٍ ما على جهاز الكمبيوتر. تُنشأ وحدات معالجة الرسومات لتنفيذ تعليماتٍ لرسومات الجرافيك على الكمبيوتر. يتطلب إنشاء رسومات الجرافيك على الكمبيوتر ومعالجتها عملياتٍ رقمية على مقاييس كبيرة؛ فالمشهد المُنشأ على الكمبيوتر هو عبارة عن مصفوفة كبيرة من الأعداد (فكِّر في الحذاء). وتعتبر وحدات معالجة الرسومات المكوِّن الأكثر فائدة وأهمية على الإطلاق في أجهزة تشغيل ألعاب الفيديو. فالتقنية نفسها التي تأسِر الذكاء البشري في ساعات التسلية واللهو تُستخدم كذلك لتطوير ذكاء الآلة.

لقد بدأنا بأبسط شبكة عصبية ممكنة؛ إذ تتكوَّن من خلية عصبية واحدة. ثم أضفنا بضع خلايا عصبية، ثم أضفنا بضع مئات أخرى. غير أن الشبكة العصبية التي أنشأناها للتعرُّف على الصور ليست كبيرة بأي حال. كما أن بنيتها ليست معقَّدة. لقد أضفنا فقط طبقة على طبقة من الخلايا العصبية. وقد حقَّق الباحثون في مجال التعلم العميق تقدُّمًا كبيرًا في ابتكار بنى الشبكات العصبية. فقد تتألَّف هذه البنى من عشرات الطبقات. ولكن لا يلزم أن تكون هندسة هذه الطبقات مجموعة من الخلايا العصبية البسيطة الأحادية الأبعاد، كالطبقات التي لدينا في المثال. على سبيل المثال، يمكن تكديس الخلايا العصبية داخل إحدى الطبقات في هياكل ثنائية الأبعاد تشبه القماشَ المُعَد للوحات الزيتية. إضافة إلى ذلك، لا يلزم أن تكون كل طبقة متصلةً اتصالًا كثيفًا بالطبقة التي قبلها؛ بل يمكن

أن تكون هناك أنماط اتصال أخرى. كذلك ليس بالضرورة أن تتصل مُخرجات طبقة بمُدخلات الطبقة التي تليها. على سبيل المثال، يمكن أن تكون هناك اتصالات بين طبقاتٍ غير متعاقبة. يمكننا تجميع الطبقات ومعاملتها كوحدات ونجمعها مع وحداتٍ تتكوَّن من طبقاتٍ أخرى لتشكيل تكويناتٍ أعقدَ وأعقدَ. وفي الوقت الحاضر، لدينا مجموعة كبيرة من بنى الشبكات العصبية، بحيث يكون هناك بنًى معيَّنة مناسبة لمهامَّ بعينِها.

تقوم الخلايا العصبية في الطبقات في جميع بنيات الشبكة العصبية بتحديث قيم الأوزان والانحياز مع التقدُّم في التعلم. إذا فكَّرنا فيما يحدث، فسنرى أن لدينا مجموعةً من المُدخلات تغيِّر الطبقات في أثناء عملية التعلم. فبمجرد أن يتوقَّف التدريب، تستوعب الطبقات بطريقةٍ ما — عن طريق التعديلات في معاملاتها — المعلومات التي تمثِّلها بيانات المُدخلات. ويمثِّل تكوين قيم الأوزان والانحياز للطبقة المُدخلات التي تتلقَّاها الطبقة. وتقوم الطبقة المخفية الأولى — المتصلة مباشرة بطبقة المُدخلات — بتحويل مُدخلات الشبكة العصبية إلى رموز. وتقوم الطبقة المخفية الثانية بتحويل مُخرجات الطبقة المخفية الأولى — المتصلة بها مباشرة — إلى رموز. وكلما تعمقنا أكثر وأكثر في شبكة متعدِّدة الطبقات، تحوِّل كل طبقة المُخرجات التي تتلقاها من الطبقة السابقة إلى رموز. فكل تمثيل يبني على ما قبله، ومِن ثَم يرتقي إلى مستوًى تجريدي أعلى من مستوى الطبقة السابقة. إذن، فالشبكات العصبية العميقة تتعلَّم سلسلةً هرمية من المفاهيم، مرتقية إلى مستوياتٍ أعلى وأعلى من التجريد. وهذا هو الإطار الذي نتحدَّث فيه عن التعلم «العميق». ونقصد بذلك بنيةً تمثِّل المستوياتُ المتعاقبة من خلالها مفاهيمَ أعمقَ بحيث تتطابق مع مستوياتِ تجريدٍ أعلى. في شبكة التعرُّف على الصور، قد تتعلم الطبقة الأولى في شبكة متعدِّدة الطبقات التعرُّف على أنماطٍ موضعية صغيرة مثل الحواف في الصور. بعد ذلك، قد تتعلم الطبقة الثانية التعرُّف على أنماطٍ مبنية من الأنماط التي تعرَّفت عليها الطبقة الأولى، مثل العيون والأنوف والآذان. كذلك قد تتعلم الطبقة الثالثة أنماطًا مبنية من الأنماط التي تعرَّفت عليها الطبقة الثانية، مثل الوجوه. يمكنك أن ترى الآن أن شبكتنا العصبية للتعرُّف على الصور كانت بسيطةً نوعًا ما؛ فلم نحاول تحقيق تعلُّم عميق بالمعنى الفعلي. ومن خلال بناء طبقات مجردة على طبقات أخرى مجردة، نتوقع من الشبكة أن تعثر على الأنماط التي يعثر عليها البشر، بدايةً من البنيات في الجمل، مرورًا بعلامات الأورام الخبيثة في صور التشخيص الطبي وصولًا إلى التعرُّف على الحروف المكتوبة بخط اليد، وكشف جرائم الاحتيال عبر الإنترنت.

غير أنك قد تقول إن كل ذلك يتلخَّص في تحديث القيم البسيطة في لبناتٍ بسيطة، ألا وهي الخلايا العصبية الاصطناعية. وستكون على صواب. وعندما يدرك الناس ذلك، يشعرون بالإحباط في بعض الأحيان. فهم يريدون أن يعلموا ماهية التعلم العميق وتعلم الآلة، ولكن بساطة الإجابة محبطة؛ إذ تكون شيئًا يبدو أنه يمكن أن يختزل قدرات البشر ويحصرها في العمليات الابتدائية الأساسية. ربما نفضِّل العثور على شيءٍ أكثر تعقيدًا لا يخفق في إشباع تقديرنا لذاتنا.

غير أنه ينبغي ألا ننسى أنه في العلم، نعتقد أنه يمكن تفسير الطبيعة استنادًا إلى المبادئ الأولى، ثم نحاول العثور على أبسط المبادئ الممكنة. وهذا لا يستبعد السلوكيات والبنى المعقَّدة المنبثقة من قواعدَ ولبناتٍ بسيطة. إن الخلايا العصبية الاصطناعية أبسطُ بكثير من نظيرتها الحيوية؛ وحتى إذا كان يمكن توضيح آليات عمل الخلايا العصبية الحيوية في نماذجَ مبسطة، فإن الفضل في هذا يرجع إلى العدد الهائل من الخلايا العصبية الحيوية المترابطة التي يمكن أن ينبع منها الذكاء، بالشكل الذي نعرفه به.

هذا يساعد على وضع بعض الأشياء في نِصابها. صحيح أن الشبكات العصبية الاصطناعية يمكن أن تكون خارقة في إمكانياتها. لكن لكي تكون ناجعة، لا بد من توافر قدْر هائل من الإبداع البشري والجهود الهندسية الجبارة. ونحن، في هذا الكتاب، لم نتطرَّق إلا إلى القشور. لنأخذ خوارزمية الانتشار العكسي على سبيل المثال. تشكِّل تلك الخوارزمية أساسَ الشبكات العصبية؛ إذ تتيح لنا الكفاءة في أداء ما يُعَد في الأساس عمليةَ إيجاد للمشتقات الرياضية. انشغل الباحثون بابتكار تقنيات حسابية فعالة، مثل «التمييز التلقائي»، وهو آلية لحساب المشتقات مستخدَمة على نطاقٍ واسع. أو لنضرب مثالًا بالطريقة الدقيقة التي تُحتسب بها التغييرات في معاملات الشبكة العصبية. طُوِّر العديد من «أدوات التحسين» ما يتيح لنا نشرَ شبكاتٍ أكبر وأكبر وفي الوقت نفسه فعَّالة أكثر وأكثر. بالنظر إلى الأجهزة، يصمِّم مهندسو الأجهزة رقاقاتٍ أفضل وأفضل لتشغيل المزيد من العمليات الحسابية العصبية على نحوٍ أسرع باستخدام إمكانيات حاسوبية أقل. وبالنظر إلى بنيات الشبكات، اقتُرحت بنيات شبكات عصبية جديدة تعمل على تحسين الشبكات الحالية. فهذا المجال مرتعٌ للأبحاث والتجارب، كما أنه ينطوي على جهودٍ لبناء شبكات عصبية تصمِّم شبكات عصبية أخرى. لذا في كل مرة ترى فيها تقريرًا إخباريًّا عن أن شبكةً عصبية حققت إنجازًا جديدًا، ارفع القبعة احترامًا للمثابرين الذين جعلوا هذا الإنجاز ممكنًا.[9]

إن الخلايا العصبية الاصطناعية أبسط بكثير من نظيرتها الحيوية، وحتى إذا كان يمكن توضيح آليات عمل الخلايا العصبية الحيوية ... فإن الفضل في هذا يرجع إلى العدد الهائل من الخلايا العصبية الحيوية المترابطة التي يمكن أن ينبع منها الذكاء.

الخاتمة

في ١٥ يوليو ٢٠١٩، قدَّم محافظ بنك إنجلترا، مارك كارني، تصميمَ الأوراق النقدية من فئة ٥٠ جنيهًا إسترلينيًّا، وتوقَّع أن تدخل التداول بعد حوالي عامين. في عام ٢٠١٨، قرَّر بنك إنجلترا الاحتفاء بشخصية علمية بطباعة صورته على الورقة النقدية الجديدة وفُتح باب الترشيح العام لمدة ستة أسابيع لاختيار هذه الشخصية. بلغ إجمالي الترشيحات ٢٢٧٢٩٩ لـ ٩٨٩ شخصية مؤهَّلة للانتخاب. ومن هنا، استقرت اللجنة الاستشارية المعنية باختيار صور الشخصيات على الأوراق النقدية على قائمةٍ قصيرة تضم ١٢ مرشحًا. بعد ذلك اتخذ محافظ البنك قراره النهائي، ووقَع الاختيار على آلان تورنج. وقال في معرض تعليقه على ذلك: «لقد كان آلان تورنج عالِم رياضيات فذًّا، وكان لعمله تأثير هائل على أسلوب معيشتنا اليوم. باعتبار آلان تورنج رائد علم الكمبيوتر والذكاء الاصطناعي، وباعتباره بطلَ حرب أيضًا، فقد كانت إسهاماته واسعة النطاق ورائدة. إن تورنج عملاق يقف على كتفيه الكثيرون الآن.»[1]

كان تورنج (١٩١٢–١٩٥٤) عبقرية فذة استكشف حدود عمليات الحوسبة وطبيعتها وتنبَّأ بظهور آلاتٍ ستُظهِر سلوكًا ذكيًّا، وتصدَّى لمسألة ما إذا كانت الآلات تستطيع التفكير، وأسهمَ كذلك في علم الأحياء الرياضية وآليات التشكل الحيوي، وكان له دور بالغ الأهمية في تحليل الرموز السرية للرسائل الألمانية المشفَّرة إبَّان الحرب العالمية الثانية (وظل إسهامه هذا طي الكتمان عقودًا). وفي منعطفٍ مأساوي للأحداث، مات تورنج منتحرًا. كان قد أُلقي القبض عليه وأُدين بالشذوذ الجنسي عام ١٩٥٢ الذي كان مُجرَّمًا في المملكة المتحدة آنذاك، ومِن ثَم أُجبر على تناول علاج هرموني. وقد صدر عفو رسمي بحقه عام ٢٠١٣. ويُعَد ظهوره على الورقة النقدية الجديدة شكلًا من إعادة الاعتبار الذي لم يكن أحدٌ ليفكِّر فيه قبل بضعة عقود.[2]

على مدى صفحات هذا الكتاب، وصفنا الخوارزميات بأنها تتكوَّن من خطوات بسيطة وسهلة لدرجة أنه يمكن تنفيذها باستخدام الورقة والقلم. وبالنظر إلى أننا ننفِّذ الخوارزميات في برامج الكمبيوتر، فإن السؤال عن ماهية الخوارزمية في الحقيقة سيساعدنا في فهم ما يمكن أن تحسبه في الواقع. وهذا يتطلب منا التعمُّق أكثر في طبيعة هذه الخطوات البسيطة. فالعمليات الحسابية التي يستطيع طالب المرحلة الابتدائية إجراءها بالورقة والقلم، في النهاية، مختلفةٌ عن العمليات التي يمكن أن يجريها خريج الجامعة. فهل يمكن أن نحدِّد بدقةٍ نوعيةَ الخطوات التي يمكن أن تتكوَّن منها الخوارزمية؟ قدَّم تورنج إجابةً لهذا السؤال حتى قبل ظهور أجهزة الكمبيوتر الرقمية. فقد قدَّم نموذجًا لآلة في عام ١٩٣٦ للإجابة على السؤال المتعلِّق بما يمكن أن يفعله أي جهاز كمبيوتر. و«آلة تورنج» عبارة عن آلة عبقرية بسيطة. وتتكوَّن من الأجزاء التالية:[3]

(١) «شريط». ينقسم الشريط إلى مربَّعات أو «خانات». كل خانة يمكن أن تكون إما فارغة أو تحتوي على رمز هجائي. ويمكن أن يكون الشريط طويلًا لما لا نهاية.

(٢) «رأس» يمكن أن يتحرك يمينًا أو يسارًا على طول الشريط، ويتحرك في اتجاه واحد في كل مرة. يمكن للرأس قراءة الرمز في الخلية التي تحته. ونطلق على الرمز في تلك الخلية «الرمز الممسوح ضوئيًّا». ويمكن للرأس أن يمحو الرمز الممسوح أو تستبدله.

(٣) «وحدة للتحكم المحدود» وتسمَّى أيضًا «مسجل الحالات». يمكن أن توجَد وحدة التحكم المحدود في أي مجموعة محدودة من الحالات. يمكنك تشبيهها بقرصٍ منقوش عليه حالات ومزوَّد بمؤشر يمكن أن يشير إلى أي حالة منها.

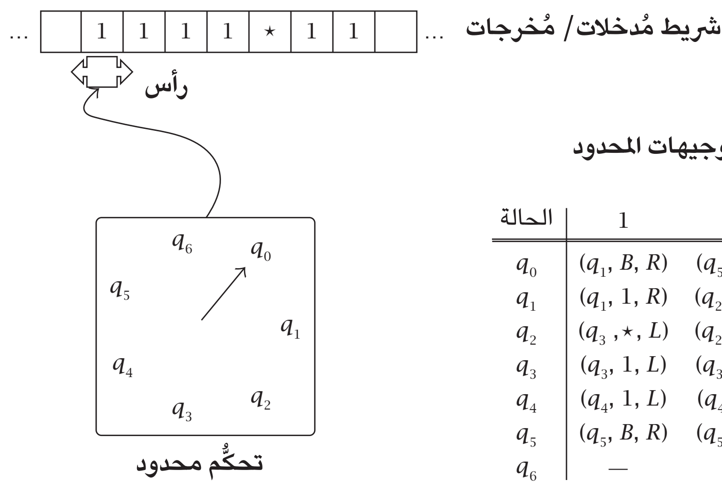
(٤) «جدول توجيهات محدود». يحدِّد كل توجيه «الحركة» التالية للآلة. هذا ما تفعله الآلة بناءً على الحالة الحالية والرمز الممسوح.

> هل يمكن أن نحدِّد بدقة نوعيةَ الخطوات التي يمكن أن تتكوَّن منها الخوارزمية؟ قدَّم تورنج ... نموذجًا لآلة في عام ١٩٣٦ للإجابة على السؤال المتعلِّق بما يمكن أن يفعله أي جهاز كمبيوتر.

يمكنك رؤية آلة تورنج في الشكل التالي.[4]

تتكوَّن الرموز الهجائية في تلك الآلة من الرقم ١ وعلامة $\star$. تُظهر وحدة التحكم المحدود أن الآلة يمكن أن تكون في حالةٍ من سبع حالات وهي q_0، q_1، …، q_6. ويحتوي

جدول التعليمات على صفٍّ لكل حالة محتملة، وعمود لكل رمز محتمَل، ونستخدم الرمز B للإشارة إلى الخانة الفارغة، حتى يمكن أن نرى ذلك الرمز. يشار إلى الحالة الحالية بالصف والرمز الممسوح ضوئيًّا بالعمود. يحتوي كل إدخال في جدول التعليمات على ثلاث قيم تصف حركة، أو قد يحتوي على شَرطة، ما يعني أن الآلة ليس لديها ما تفعله في هذه المجموعة الثنائية من الصف والعمود.



جدول التوجيهات المحدود

الحالة	رمز 1	⋆	B
q_0	(q_1, B, R)	(q_5, B, R)	—
q_1	$(q_1, 1, R)$	$(q_2, \star, R)$	—
q_2	$(q_3, \star, L)$	$(q_2, \star, R)$	(q_4, B, L)
q_3	$(q_3, 1, L)$	$(q_3, \star, L)$	(q_0, B, R)
q_4	$(q_4, 1, L)$	(q_4, B, L)	$(q_6, 1, R)$
q_5	(q_5, B, R)	(q_5, B, R)	(q_6, B, R)
q_6	—	—	—

تتكوَّن حركة الآلة من ثلاثة إجراءات:

(١) قد تغيِّر الآلةُ الحالةَ الحالية أو تبقى عليها. الحالة الجديدة هي العنصر الأول من القيم الثلاث في جدول التعليمات المحدود.

(٢) ستكتب الآلة رمزًا أسفل الرأس. قد يتطابق الرمز مع الرمز الموجود بالفعل (وفي هذه الحالة يبقى الرمز الحالي في الخانة). والرمز المراد كتابته هو العنصر الثاني في القيم الثلاث.

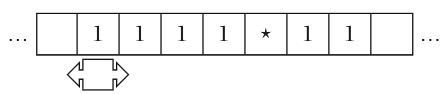
(٣) سيتحرك الرأس إما يسار الخانة الحالية (L) أو يمينها (R). ويكون الانتقال هو العنصر الثالث في القيم الثلاث.

يطبِّق مثال آلة تورنج خوارزميةً تحسب الفرق بين العددين a وb حين تكون $a > b$؛ وإلا فسيكون الناتج صفرًا. تسمَّى هذه العملية «الطرح المبتور» ويُكتب بالصيغة $a \dot{-} b$. وبذلك يصبح لدينا $4 \dot{-} 2 = 2$ و$2 \dot{-} 4 = 0$.

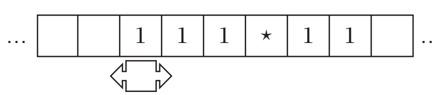
في البداية، نضع «مدخلات» الآلة على الشريط. والمُدخلات هنا عبارة عن سلسلة محدودة من الرموز مأخوذة من هجائية الآلة. بذلك تكون جميع خانات الشريط، الواقعة يمينه ويساره، فارغة. والمُدخلات في هذا النموذج لآلة تورنج هي ١١١١ ⋆ ١١. تعبِّر المُدخلات عن العددين أربعة واثنين في «نظام العد الأحادي»، يُفصل بينهما بالرمز ⋆.

حين تبدأ هذه الآلة عملَها يكون الرأس على خانة المُدخلات أقصى اليسار. تشير وحدة التحكم المحدود إلى الحالة q_0. ثم تبدأ الآلة في العمل وتنفيذ تحرُّكاتها. إذا تتبَّعنا عمل الآلة في أول ست حركات، فسنرى أنها تسير بالنمط التالي:

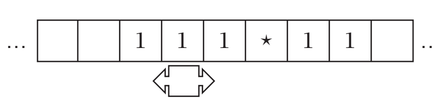
(١) الآلة في الحالة q_0 والرمز الممسوح هو ١:



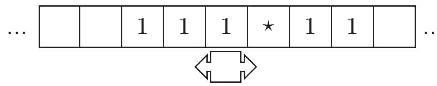
يعطينا جدول التوجيهات (q_1، B، R)، ولذا ستغير الآلة الحالة إلى q_1 وستبدل خانة الرقم ١ إلى خانة فارغة وتتحرك جهة اليمين. سيصبح الشريط والرأس كما يلي:



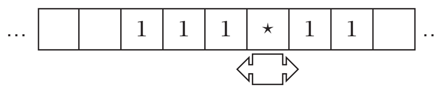
(٢) بالنسبة إلى الحالة q_1 والرمز الممسوح ١، يعطينا جدول التوجيهات (q_1، 1، R). ستقرأ الآلة القيمة ١ وتكتبها وتترك الخانة كما هي وستتحرك جهة اليمين، بحيث تبقى عند الحالة q_1:



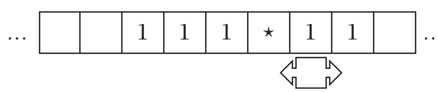
(٣) تفعل الآلة كما في الخطوة ٢، حيث تقرأ القيمة ١ وتكتبها وتبقى عند الحالة q_1، وتتحرك جهة اليمين:



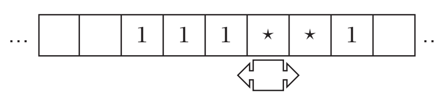
(٤) مرة أخرى، ستقرأ الآلة القيمة ١ وتكتبها وتبقى عند الحالة q_1 وتتحرك جهة اليمين:



(٥) تحرَّك الرأس تاركًا الرمز ⋆ وبقي عند الحالة q_1. التوجيهات هي (q_2، ⋆، R). ستغيِّر الآلة حالتها إلى q_2 وتترك الرمز ⋆ على الشريط وتتحرك جهة اليمين:



(٦) تحرَّك الرأس مغادرًا الرمز ١ الواقع يمين الرمز ⋆ وأصبح عند الحالة q_2. التوجيهات هي (q_3، ⋆، L). ستغيِّر الآلة حالتها إلى q_3 وتبدِّل الرمز ⋆ مكان الرمز ١ وتعود يسارًا:



ستستمر الآلة في العمل على هذه الوتيرة بحيث تؤدي التحركات التي يُمليها جدول التوجيهات. إذا نظرنا من مستوًى أعلى، فسندرك أن الآلة تنفِّذ حلقة. ففي كل تكرار، تجد الرمز ١ أقصى اليسار وتبدله إلى خانة فارغة. بعد ذلك تبحث جهة اليمين عن الرمز ⋆. وعندما تجده، تستمر في التحرُّك جهة اليمين حتى تجد الرمز ١ وتحوِّله إلى ⋆. لذا في كل

تكرار، تشطب الآلة أحد الرموز ١ يمين ويسار الرمز ⋆. وعند نقطة محدَّدة، لن يعود ذلك ممكنًا. عندئذٍ، ستبدِّل الآلة كل رموز ⋆ إلى خاناتٍ فارغة وتنهي عملها. سيحتوي الشريط على العدد ١١ المكافئ للعدد ٢ محاطًا بخانات فارغة. وللإشارة إلى إنهاء العمل، تدخل الآلة الحالة q_6، حيث لا يوجد ما تفعله حسب جدول التوجيهات، وعندئذٍ ستتوقف.

إذا زودنا الآلة بمُدخلات في صورة ١١ ⋆ ١١١١، فستعمل الآلة بكامل طاقتها إلى أن تتوقَّف وقد امتلأ الشريط بالخانات الفارغة، وتلك الخانات تساوي صفرًا. إذا أعطينا الآلة أي مدخلات تتكون من a من العدد واحد متبوعة بالعلامة النجمية ثم b من العدد واحد، فستستمر في حركتها إلى أن تترك الشريط إما بالقيمة $a - b$ من العدد واحد — إذا كانت $a > b$ — أو ستترك الخانات كلها فارغة.

تستخدم آلة تورنج هذه خوارزميةً لحساب عملية الطرح المبتور بناءً على المُدخلات واتباع التعليمات التي يمليها جدول التوجيهات. الخطوات أولية للغاية لدرجةٍ أن رأس آلة تورنج يتنقل كثيرًا من أجل إجراء العملية. سيستغرق ٢١ حركة لإيجاد أن $2 \dot{-} 4 = 0$، و34 حركة لإيجاد أن $4 \dot{-} 2 = 2$. ولكن ما أبسط تلك التحركات! بإمكان أي شخص متوسط الذكاء أن ينفِّذها. وهذه الطبيعة البدائية للخطوات هي نفسها السر. فلست بحاجة إلى مؤهلاتٍ متقدمة لتنفيذ خطوات آلة تورنج؛ كلُّ ما تحتاجه هو الاطلاع على الجدول والتحرك حول شريط وقراءة رمز واحد وكتابته في المرة الواحدة، وتتبُّع مسار الحالة. هذا كل ما في الأمر. ولكن الآلة ليست تافهة؛ لأن الإجابة على السؤال الخاص بنوعية الخطوات التي يمكن أن تتألف منها خوارزميةٌ ما، هي الخطوات التي يمكن أن تنفِّذها آلة تورنج.

في هذا الكتاب، تناولنا الخوارزميات بمستوًى أعلى وبخطواتٍ أعقدَ. وهذا مريح لنا؛ لأن آلة تورنج تعمل بمستوًى من التفصيل المحدود يجعل استخدامها لوصف الخوارزميات التي تناولناها غير عملي تمامًا. لكن كل الخطوات التي تنطوي عليها الخوارزميات التي تناولناها يمكن عرضها كخطوات لنموذجٍ منشأ على نحوٍ صحيح لآلة تورنج. لقد وصفنا نموذجًا بسيطًا لآلة تورنج لتنفيذ عملية الطرح المبتور. لتنفيذ خوارزميةٍ أعقدَ، كنا سنحتاج إلى نموذج لآلة تورنج ذي حالات أكثر ورموز هجائية أكبر وجدول توجيهات أكبر. لكن ظل بإمكاننا بناء ذلك النموذج لو شئنا.

إن بساطة آلة تورنج لا تعبِّر عن حدود إمكانياتها؛ فبناءً على أي خوارزمية، يمكننا بناء آلة تورنج تنفِّذ تلك الخوارزمية. لما كانت أجهزة الكمبيوتر تشغل الخوارزميات، فإن

أي خوارزمية يستطيع الكمبيوتر حسابها، يمكن أن تحسبها آلة تورنج. بعبارة أخرى، «كلُّ ما نستطيع فعله باستخدام الخوارزميات، يمكن أن نفعله باستخدام آلة تورنج». هذا عرض مسهب لـ «أطروحة تشيرش- تورنج»، التي سمِّيت نسبةً لتورنج وعالِم الرياضيات الأمريكي ألونزو تشيرش (١٩٠٣–١٩٩٥)، أحد مؤسسي علم الكمبيوتر النظري. وبما أنها «أطروحة»، فهي ليست شيئًا ثبتت صحته، ولا نعرف هل يمكن إثباتها رياضيًّا أم لا. من المحتمل، من الناحية النظرية، أنه يمكن عدم إثباتها إن اخترع شخصٌ ما شكلًا بديلًا للحوسبة يحسب الأشياء التي يتعذَّر على آلة تورنج حسابها. ولا نظن أن ذلك شيء وارد حدوثه. ولذا نعتبر آلة تورنج وصفًا رسميًّا لفكرة الخوارزميات.[5]

يمكنك تخيُّل أي كمبيوتر بالقدرات التي تريدها. سيصبح الكمبيوتر أسرعَ بكثير من آلة تورنج التي تعمل على شريطٍ من الرموز كما ذكرنا. ولكن أي شيء يحسبه الكمبيوتر بالخوارزميات، تستطيع آلة تورنج حسابه أيضًا. يمكنك حتى أن تتخيَّل أجهزة كمبيوتر لم نخترعها حتى الآن. تعمل أجهزة الكمبيوتر بوحدات «البت» التي لا توجد إلا بقيمتين هما: ٠ و١. أما «أجهزة الكمبيوتر الكمية» فتعمل بوحدات «الكيوبت». عندما نفحص حالة وحدات الكيوبت، سنجدها مكوَّنة من القيمة ٠ أو ١ مثل وحدات البت. لكن عندما لا نفحص وحدات الكيوبت، فيمكن أن تكون في الحالتين الثنائيتين ٠ و١ فيما يسمَّى بـ «التراكب». يبدو الأمر كما لو كانت وحدات الكيوبت تتكوَّن من كلٍّ من ٠ و١ إلى أن نقرِّر قراءتها، وعندها تقرِّر أن تصبح قيمة من هاتين القيمتين. وهذا يتيح لأجهزة الكمبيوتر الكمية أن تعبِّر عن عدة حالات من الحوسبة في وقتٍ واحد. وسيتيح لنا جهاز الكمبيوتر الكمي حلَّ المسائل الحسابية السريعة التي لا يسهُل على أجهزة الكمبيوتر التقليدية حلها. لكن للأسف، بناءُ كمبيوتر كميٍّ صعبٌ في ظل التكنولوجيا الحالية. وحتى الكمبيوتر الكمي لا يستطيع إنجاز مهامَّ لا تستطيع آلة تورنج إنجازها. وعلى الرغم من أن بإمكانه حل بعض المسائل بكفاءة تفوق أي كمبيوتر عادي في الوقت الحالي أو أي آلة تورنج، فإنه لن يستطيع حل أي مسائل لا تستطيع آلة تورنج حلها.

تتجسَّد قيود الحوسبة في آلات تورنج. فأي شيء يمكن للكمبيوتر القيام به يمكننا إنجازه بالورقة والقلم بالعمل على شريط من الرموز. وكل شيء تراه يُنفَّذ على أي جهاز رقمي هو في الواقع سلسلة من عمليات المعالجة الأولية للرموز. في علوم الطبيعة، ننظر إلى الكون ونعتقد أن بمقدورنا تفسير نواميسه باستخدام المبادئ الأساسية. أما في الحوسبة، فالعكس هو الصحيح. فنحن نمتلك المبادئ الأساسية ونعتقد أنه يمكننا تحقيق نجاحات باهرة باستخدامها.

عندما طرح تورنج آلته باعتبارها نموذجًا لحوسبة، لَّما تكن أجهزة الكمبيوتر الرقمية قد ظهرت بعد. وهذا لم يمنعه من استكشاف إمكانيات أجهزة الحوسبة التي ستُصنع في المستقبل. عندما نفكِّر في حدود أجهزة الكمبيوتر، ينبغي أن نتذكَّر أيضًا العجائب التي أنشأها العقل البشري داخل هذه الحدود. فحدود الحوسبة لم تنتقص من إبداعنا لمواصلة تطوير خوارزميات تصلح لكل مناحي الحياة. عندما اختُرعت الكتابة في بلادِ ما بين النهرين، كان الغرض منها المساعدة في حفظ السجلات، وليس كتابة الأعمال الأدبية. ربما كان الكتَّاب الأوائل محاسِبين، وليس مؤلِّفين، ولكن من تلك البدايات المتواضعة برز ويليام شكسبير. ومن يعلم ما يمكن أن تخرجه الخوارزميات في المستقبل.

> تتجسَّد قيود الحوسبة في آلات تورنج. فأي شيء يمكن للكمبيوتر القيام به يمكننا إنجازه بالورقة والقلم ... وكل شيء تراه يُنفَّذ على أي جهاز رقمي هو ... سلسلة من عمليات المعالجة الأولية للرموز.

مَسْرَد المصطلحات

اتصال مكثَّف: ترتيب الطبقات في الشبكة العصبية بحيث تتصل جميع الخلايا العصبية في طبقةٍ ما بكل الخلايا العصبية في الطبقة التالية.

ارتباط تشعُّبي: مرجع من نص إلى جزء آخر في النص أو إلى نص مختلف. على الشبكة العنكبوتية، الارتباطات التشعُّبية عبارة عن روابط بين صفحات الويب التي قد يتبعها المستخدم في أثناء تصفُّحه.

استدلال: استراتيجية للاختيار من بين عدة بدائل في خوارزمية. سيتطلب منَّا الاستدلال الجشع أن نأخذ الخيار الذي يبدو أنه الأفضل في الوقت الحالي (بصرف النظر عمَّا قد يحدث في المستقبل).

أطروحة تشيرش-تورنج: فرضية تنصُّ على أن أي شيء يمكن حسابه باستخدام خوارزمية، يمكن حسابه باستخدام آلة تورنج.

أقصر: أقصر مسار بين عقدتين في التمثيل البياني.

آلة الجدولة: أجهزة كهروميكانيكية تستطيع قراءة البطاقات المثقبة وتستخدم المعلومات الواردة بها في إجراء عملية إحصائية.

آلة تورنج: آلة مثالية (تجريدية) اخترعها آلان تورنج وتتكوَّن من شريطٍ لا نهائي ورأس متحرك يقرأ الرموز على الشريط ويكتبها باتباع مجموعة من القواعد المحدَّدة مسبقًا. يمكن لآلة تورنج أن تنفِّذ أي خوارزمية، ومِن ثَم يمكن استخدامها باعتبارها نموذجًا للأشياء التي يمكن حوسبتها.

إنترنت: شبكة عالمية من أجهزة الكمبيوتر والأجهزة الرقمية تتصل فيما بينها عن طريق مجموعة مشتركة من بروتوكولات الاتصال. في البداية، كان أول حرف فيها يُكتب كبيرًا في الإنجليزية (Internet) لأن كلمة internet كان يمكن أن تشير إلى أي شبكةٍ تمتد إلى ما وراء الحدود الداخلية للمؤسسة، والتي تسمَّى الإنترانت. لكن مع انطلاق شبكة الإنترنت العالمية، لم تَعُد كتابة الحرف الأول كبيرًا مفضلة، ما أدَّى في الغالب إلى توفير كمية كبيرة من الحبر.

إنتروبيا متقطعة فئوية: دالة خسارة تَحسُب الفرق بين توزيعين للاحتمالية.

انتقاء: في الخوارزميات والبرمجة، هو اختيار بين سلسلة من الخطوات البديلة لتنفيذها، بناءً على حالة منطقية.

انحياز: قيمة عددية تُرفق بالخلية العصبية تتحكَّم في نزوعها إلى التحفيز.

بادئة: الجزء المنطوق في إيقاعٍ ما.

بِت: الوحدة الأساسية للمعلومات المخزَّنة على الكمبيوتر. قد تأخذ وحدة البت إحدى القيمتين ٠ أو ١. كلمة بت في الإنجليزية (bit) مشتقة من مصطلح binary digit بمعنى الرقم الثنائي.

بحث ثنائي: خوارزمية بحث تعمل على البيانات المرتَّبة. نتحقَّق من العنصر في وسط مساحة البحث. إذا تطابق مع العنصر الذي نبحث عنه، فهذا جيد. وإن لم يتطابق، نكرِّر الإجراء جهة النصف الأيسر أو الأيمن بناءً على إذا ما كنا قد تجاوزنا المستهدف أم لم نتجاوزه.

بحث خطي: خوارزمية بحث نفحص فيها كل عنصر تِباعًا إلى أن نجد العنصر الذي نبحث عنه. يطلق عليه أيضًا البحث التسلسلي.

بحث ذاتي التنظيم: خوارزميات بحث تستخدم شهرة العناصر قيد البحث عن طريق نقلها إلى أماكنَ حيث نستطيع العثور عليها على نحوٍ أسرع.

برمجة: فن كتابة برامج الكمبيوتر.

برمجيات: مجموعة البرامج التي تعمل على جهاز كمبيوتر أو جهاز رقمي؛ المصطلح مكمِّل لمصطلح مكونات الأجهزة. استُخدم المصطلحان قبل أجهزة الكمبيوتر في العديد من المواقع. ففي عام ١٨٥٠، كان جامعو القمامة يستخدمون المصطلحين للتمييز بين

المواد التي تتحلل وبين غيرها من المواد الأخرى. هذان المعنيان قد يواسيان أيَّ شخص يعاني مع جهاز كمبيوتر لا ينجز المهام التي يفترض أن ينجزها.

برنامج: مجموعة التعليمات مكتوبة بلغة برمجة وتصِف عملية حاسوبية.

بطاقة مثقبة: قطعة من الورق المقوى تسجِّل المعلومات حسب موقع الثقوب فيها. يطلق عليه أيضًا بطاقة الثقوب. استُخدمت تلك البطاقات في أجهزة الكمبيوتر الأولى، واستُخدمت قبلها في آلاتٍ مثل مناسج جاكارد للنسيج؛ إذ كانت تصِف التصميم الذي سيُنسج.

بنية البيانات: طريقة لتنظيم البيانات بحيث يمكننا التعامل مع البيانات باستخدام مجموعةٍ محدَّدة وموصوفة من العمليات.

بنية التحكُّم: الطرق الثلاث التي يمكن من خلالها تجميع خطوات في خوارزمية أو برنامج وهي: التسلسل والانتقاء والتكرار.

بيانات قابلة للفصل خطيًّا: مجموعة بيانات يمكن فصل ملاحظاتها إلى فئتين باستخدام خط مستقيم في بُعدين، أو باستخدام المستوى في ثلاثة أبعاد، أو باستخدام المستوى الفائق في مزيد من الأبعاد.

بيرسيبترون: خلية عصبية اصطناعية تستخدم الدالة الدرجية لتنشيطها.

تأثير ماثيو (أو متى): ظاهرة زيادة غنى الغني وزيادة فقر الفقير. الاسم مشتق من إنجيل متَّى (٢٥: ٢٩) ووُجد أنه ينطبق على العديد من السياقات وليس على الثروة المادية فقط.

تبديل: إعادة تنظيم بعض البيانات بترتيب مختلف.

تحفيز (الخلية العصبية): انظر التنشيط (الخلية العصبية).

تخفيف: طريقة في خوارزميات التمثيل البياني نعيِّن فيها أسوأ قيمة ممكنة للقيم التي نريد إيجادها، وتتقدَّم الخوارزمية بتقديم تقديرات أفضل وأفضل لهذه القيم. ولذا نبدأ بأبعد القيم المحتملة ونخففها تدريجيًّا بقيمٍ أقربَ وأقرب إلى النتيجة النهائية.

تدرُّج: متجه يتضمَّن كل المشتقات الجزئية للدالة.

تدريب: في تعلُّم الآلة، هي عملية نغذي فيها الخوارزمية بمدخلات أمثلة بحيث يمكنها أن تتعلم تقديم مخرجات صحيحة.

ترتيب انتقائي: طريقة ترتيب نعثر خلالها في كل مرة على أصغرِ العناصر غيرِ المرتَّبة ونضعها في مواضعها الصحيحة.

ترتيب بالإدراج: طريقة ترتيب نأخذ فيها كل عنصر على حدةٍ وندرجه في موضعه الصحيح بين العناصر المرتَّبة بالفعل.

ترتيب بالجذر: طريقة ترتيب تعمل بتقسيم المفاتيح إلى أجزائها المكونة (مثل تقسيم الأعداد إلى مفاتيح رقمية)، ووضع العناصر في أكوام مناظرة لقيم أجزائها (مثل عشرة أكوام، كومة لكل رقم). نبدأ بتكوين الأكوام بناءً على الرقم الأخير، ثم نجمع كل الأكوام ونعيد توزيعها على الأكوام بناء على الكومة قبل الأخيرة وهكذا. عندما ننفِّذ الإجراء للعدد الأول، ينتهي بنا الحال بكومةٍ مرتَّبة. إنها طريقة فرز حسب السلسلة لأننا نعامل المفاتيح الرقمية كسلسلة من الأعداد.

ترتيب بالدَّمج: طريقة للترتيب تعمل بتكرار الدَّمج بين مجموعاتٍ أكبرَ وأكبر من العناصر المرتَّبة.

ترتيب سريع: طريقة ترتيب تعمل بتكرار اختيار عنصر وتحريك العناصر الأخرى من حوله بحيث تصبح جميع العناصر الأخرى الأصغر على أحد جانبيه وباقي العناصر على جانبه الآخر.

ترميز Big O: ترميز للتعبير عن التعقيد الحسابي. عند وجود خوارزمية ومُدخلاتها أكبر من حدٍّ معيَّن، يعطينا هذا الترميز حدًّا أعلى على عدد الخطوات المتوقَّع الذي تتطلبه الخوارزمية كي تكتمل. نريد أن تكون المُدخلات أكبر من حدٍّ معيَّن لأننا معنيون بسلوك الخوارزمية في البيانات الضخمة. يضمن لنا تعقيد ترميز Big O لأي خوارزمية ألا تتطلبَ هذه الخوارزمية أكثرَ من عددِ خطواتٍ معيَّن بالنسبة إلى البيانات الضخمة. على سبيل المثال، التعقيد $O(n)^2$ يعني أنه بالنسبة إلى مُدخلات بحجم n التي تتجاوز حدًّا معينًا، لن تتطلب الخوارزمية أكثرَ من المضاعف الثابت للعدد n^2 من الخطوات حتى تكتمل.

تَسلسُل: سلسلة من الخطوات تنفَّذ واحدة تلو الأخرى في الخوارزميات والبرمجة.

تسلُّق التل: استعارة للتعبير عن حلِّ المسائل. يوجد الحل على قمَّة التل وينبغي أن نصعد من عند السفح. وفي كل خطوة، قد يكون علينا اتخاذ قرار بشأن المسار الذي نسلكه من بين المسارات البديلة. وبناءً على اختياراتنا، قد نختار المسار الأفضل بوجه عام، قد

لا يكون هو المسار الأفضل ولكنه يأخذنا إلى القمة، أو للأسف قد نختار مسارًا يؤدي إلى مرتفع. وإذا وقع الأسوأ ووصلنا إلى مرتفع، فسنضطر إلى العودة إلى موضعٍ سابق لنتخذ مسارًا جديدًا تمامًا.

تسمية: في تعلُّم الآلة، هي قيمة تعبِّر عن الفئة التي تندرج تحتها الملاحظة. في مرحلة التدريب، يعطى الكمبيوتر مسائلَ مرفقةً بالحلول؛ عندما تكون المسألة عبارة عن مسألة تصنيف، تكون الحلول هي التسميات التي تعبِّر عن الفئات.

تشابك عصبي: رابطة تصل بين الخلايا العصبية.

التشظي: انقسام المادة إلى أجزاء أصغر. والمادة، في الفيزياء النووية، عبارة عن نواة ثقيلة تصدر عددًا هائلًا من البروتونات والنيوترونات بعد قصفها باستخدام جسيم ذي طاقة عالية.

التعرُّف على الصور: مهمة حوسبية تتمثَّل في التعرُّف على الأنماط في الصور.

تعقيد (تعقيد حسابي): الوقت الذي تتطلبه الخوارزمية كي تعمل. يعبَّر عن الوقت بترتيب الخطوات الحسابية الأولية المطلوب لإكمال المهمة.

تعقيد المضروب: تعقيد حسابي يتبع نمو المضروب. في ترميز Big O، تكون الصيغة $O(n!)$.

تعلُّم الآلة: استخدام الخوارزميات التي تحل المسائل عن طريق التعليم التلقائي من الأمثلة.

تعلُّم عميق: الشبكات العصبية التي تتكوَّن من العديد من الطبقات المخفية، تُنظَّم بحيث تعبِّر الطبقات اللاحقة عن مفاهيمَ أعمقَ وتتطابق مع مستوياتِ تجريدٍ أعلى.

تعلُّم غير موجَّه: نهج في تعلُّم الآلة نزود فيه خوارزمية بمسائل المُدخلات بدون الحلول. عندئذٍ، لا بد لخوارزمية تعلُّم الآلة أن تستنبط المُدخلات المتوقَّعة لكي تتمكَّن من تقديم المخرجات.

تعلُّم موجَّه: نهج في تعلُّم الآلة نزوِّد فيه خوارزمية بمسائل المدخلات مصحوبة بالحلول.

التقريب: حل مسألةٍ باستخدام خوارزمية قد لا تجد الحل الأمثل، ولكنها قد تجد حلًّا ليس بعيدًا عنه.

تكرار: انظر الحلقة.

تلوين التمثيل البياني: تلوين الحواف أو الرءوس في التمثيل البياني.

تلوين الحواف: تعيين ألوانٍ لحواف التمثيل البياني بحيث لا تتشارك حافتان متجاورتان اللونَ نفسه.

تلوين الرءوس: تعيينُ ألوانٍ لرءوسِ تمثيل بياني بحيث لا يتشارك رأسان متجاوران لونًا واحدًا.

التمثيل البياني غير الدوري: تمثيل بياني ليس له دورة.

تمثيل بياني: مجموعة من العُقَد — تسمَّى أيضًا الرءوس — والحواف — وتسمَّى أيضًا الروابط — التي تصل بين هذه العُقَد. يمكن استخدام التمثيلات البيانية لتمثيل أي نوع من البنى المترابطة، بدايةً من الأشخاص وحتى شبكات الكمبيوتر. ونتيجة لذلك، يمكن تمثيل العديد من المسائل في شكل رسوم بيانية، وقد طُوِّر العديد من الخوارزميات التي تعمل إلى جانب تلك التمثيلات البيانية.

تمثيل بياني غير موجَّه: تمثيل بياني تكون الحواف فيه غير موجَّهة.

تمثيل بياني متعدِّد: تمثيل بياني يمكن أن تظهر فيه إحدى الحواف أكثرَ من مرة.

تمثيل بياني موجَّه: تمثيل بياني تكون الحواف فيه موجَّهة.

تمييز تلقائي: مجموعة تقنيات لتقييم مشتقة دالةٍ ما عدديًّا؛ أي ليس تحليليًّا، ما يستتبع استخدام قواعد التفاضل والتكامل لتمييز الدوال.

جولة: مسار يبدأ وينتهي عند العقدة نفسها في التمثيل البياني. يُطلق عليها أيضًا اسم الدورة.

جولة أويلرية: مسار أويلري يبدأ وينتهي عند العقدة نفسها. يُطلَق على ذلك المسار جولة أويلرية أيضًا.

حد القرار: قيم لمتغير أو أكثر تشكِّل الحد الفاصل بين نتيجتين مختلفتين لقرار واحد بناءً على المتغير أو المتغيرات.

الحل الأمثل الشامل: أفضلُ حلٍّ للمسألة بوجه عام.

الحل الأمثل الموضعي: حل أفضل من جميع الحلول الأخرى المجاورة، ولكنه ليس الأفضل بوجه عام. الحل المجاور هو حل يمكننا الوصول إليه بالتحرك خطوة واحدة من الحل الذي بين أيدينا.

حلقة: سلسلة متتابعة من التعليمات تتكرَّر في برنامج كمبيوتر. تنتهي الحلقة عند استيفاء أحد الشروط. أما الحلقة التي لا تنتهي فهي عبارة عن حلقة لا نهائية، وعادة ما تكون خطأً برمجيًّا لأنها قد تقود البرنامج إلى الإخفاق في التوقف. انظر التكرار.

خسارة: الفرق بين المُخرجات الفعلية والمخرجات المطلوبة في خوارزمية من خوارزميات تعلُّم الآلة. وعادة ما تُحسب بواسطة دالة خسارة.

خطأ برمجي (bug): خطأ في برنامجٍ ما. استُخدم لفظ bug (بمعنى حشرة) من قِبل توماس أديسون للتعبير عن وجود عيب تقني. في بداية ظهور الحوسبة، كانت هناك حشرات حقيقية عرفت طريقها إلى الأجهزةِ ما تسبَّب في تعطيلها. وقد وُجدت حشرة عُثَّة فعلت ذلك في جهاز الكمبيوتر هارفارد مارك ٢ عام ١٩٤٧. حُفظت الحشرة في سجل الجهاز الذي يعتبر جزءًا من المجموعة الموجودة في متحف سميثسونيان الوطني للتاريخ الأمريكي.

خلية عصبية: الخلية العصبية هي خلية تشكِّل اللبنة الأساسية للجهاز العصبي. تتلقى الخلية الإشارات من الخلايا العصبية الأخرى وتنشرها إلى الخلايا العصبية الأخرى في الجهاز العصبي.

خوارزميات التحسين: خوارزميات تحسِّن قيمة الدالة إلى أقصى حدٍّ ممكن. في تعلُّم الآلة، عادة ما تقلل خوارزميات التحسين قيمة دالة الخسارة إلى أدنى حد.

خوارزمية:

(١) اذهب إلى الصفحة الأولى من الكتاب.
(٢) اقرأ الصفحة الحالية.
(٣) إذا لم تفهم، انتقل إلى الخطوة ٢. ولو لم تُفِدك، انتقل إلى الخطوة رقم ٤.
(٤) إذا كانت هناك صفحة تالية، اجعلها صفحتك الحالية وانتقل إلى الخطوة ٢. أما لو لم يكن هناك صفحة، فتوقَّف.

خوارزمية إقليدس: خوارزمية لإيجاد العامل المشترك الأكبر لعددين صحيحين، وردت في كتاب «الأصول»، وهي مجموعة تضم ١٣ كتابًا من تأليف عالِم الرياضيات اليوناني القديم إقليدس (حوالي سنة ٣٠٠ قبل الميلاد). يتناول كتاب «الأصول» الهندسة ونظرية الأعداد حيث يبدأ بالبديهيات وإثبات المبرهنات بناءً على البديهيات. يعتبر هذا الكتاب

من أقدم المؤلَّفات التي لا تزال باقية في الرياضيات التي تستخدم هذا النهج الاستدلالي، ومِن ثَم فهو من أكثر الكتب تأثيرًا في تاريخ العلم.

خوارزمية الانتشار العكسي: خوارزمية أساسية لتدريب الشبكات العصبية. تصحِّح الشبكة تكوينها (أي أوزانها وانحيازاتها) عن طريق نشر تعديلاتٍ بدايةً من الطبقة الأخيرة وصولًا إلى الطبقة الأولى.

خوارزمية بيج رانك: خوارزمية تُستخدم لترتيب صفحات الويب من حيث أهميتها. طوَّر هذه الخوارزمية مؤسِّسا شركة جوجل وكانت أساس محرك البحث جوجل. رتبة صفحة الويب هي ترتيبها بين الصفحات.

خوارزمية جشعة: خوارزمية نستخدمها عندما نضطر للاختيار بين مسارات عمل بديلة، وفيها نختار المسار الذي يعطينا أفضل نتيجة فورية. وليس بالضرورة أن يؤدي ذلك إلى الناتج الأمثل في النهاية.

خوارزمية ديكسترا: خوارزميةٌ اخترعها عالِم الكمبيوتر الهولندي إدسخر ديكسترا في عام ١٩٥٦ لإيجاد أقصر مسار بين عقدتين في تمثيل بياني. تصلح تلك الخوارزمية مع التمثيلات البيانية التي تتضمن قيمَ أوزان موجبة.

خوارزمية فورية: خوارزمية لا تحتاج إلى كل مُدخلات المسألة كي تعطينا حلًّا. تحصل الخوارزمية الفورية على المُدخلات بالتدريج، مع وصول هذه المُدخلات، وعند كل مرحلة تقدم حلًّا يأخذ في الاعتبار المدخلات التي تلقتها حتى الآن.

خوارزمية هيرهولزر: خوارزمية لإيجاد دورات أويلر في التمثيلات البيانية. وقد نشرها عالِم الرياضيات الألماني كارل هيرهولزر في عام ١٨٧٣.

دالة الظل الزائدي: إحدى دوال التنشيط التي تشبه الدالة السينية، ولكن مخرجاتها تتراوح بين −١ و١.

دالة تنشيط: دالة تحدِّد مُخرجات الخلية العصبية بناءً على مُدخلاتها.

دالة سوفت ماكس: إحدى دوال التنشيط التي تأخذ المُدخلات في صورة متجه للأعداد الحقيقية وتحوِّلها إلى متجه آخر يعبِّر عن توزيع للاحتمالية.

دالة سينية: دالة على شكل حرف S تتراوح قيمها بين ٠ و١.

دالة مصحح: إحدى دالات التنشيط التي تحوِّل كلَّ المُدخلات السالبة إلى صفر، وإلا تناسبت مُخرجاتها مباشرةً مع مُدخلاتها.

درجة (عقدة): عدد الحواف المتاخمة لعقدةٍ ما.

دورة: في التمثيلات البيانية، هي المسار الذي يبدأ وينتهي عند العقدة نفسها.

رابط خلفي: رابط يشير إلى صفحة ويب نزورها، وبالتبعية صفحات الويب التي تحتوي على روابط تشير إلى صفحة الويب التي نزورها.

رأس: أول عنصر في قائمةٍ ما.

سِجل: مجموعة من البيانات المترابطة تصِف كيانًا لتطبيقٍ معيَّن. على سبيل المثال، يمكن أن يتضمن سجل الطالب بيانات الهوية وسنة الالتحاق والدرجات.

سلسلة: تسلسُل من الرموز. قديمًا، كانت السلسلة عبارة عن تسلسل من الحروف، ولكن في الوقت الحاضر يعتمد ما يندرج في سلسلةٍ على التطبيق الفعلي؛ فقد يكون أرقامًا أو حروفًا أو علامات ترقيم أو حتى الرموز التي اختُرعت حديثًا مثل الرموز التعبيرية.

شبكة اجتماعية: تمثيل بياني تعبِّر العُقَد فيه عن الأفراد وتعبِّر الحواف عن العلاقات بينهم.

طبقة مخفية: طبقة في الشبكة العصبية ليست متصلة بمدخلات الشبكة أو مخرجاتها اتصالًا مباشرًا.

طريقة الأُس: خوارزمية تبدأ بمتجه وتضربه في مصفوفة، ثم تكرر عمليات الضرب في المصفوفة حتى تتلاقى عند قيمة ثابتة. طريقة الأُس هي صميم خوارزمية بيج رانك؛ والمتجه الذي تتلاقى عنده هو المتجه الذاتي الأول في مصفوفة جوجل.

طريقة الترتيب بالسلسلة: طريقة ترتيب تعامل مفاتيحها باعتبارها تسلسلًا من الرموز. على سبيل المثال، المفتاح ١٢٣٤ يُعامَل كسلسلة من الرموز ١، ٢، ٣، ٤ بدلًا من العدد ١٢٣٤.

طريقة تبديل الموضع: خوارزمية بحث ذاتية التنظيم. عندما نعثر على عنصر، نبدله مع العنصر الذي قبله. بهذه الطريقة، تتحرك العناصر الشائعة إلى المقدمة.

طول المسار: مجموع قيم الأوزان عبر مسارٍ ما في التمثيل البياني. إذا لم يتضمَّن التمثيل البياني أوزانًا، يكون طول المسار هو عدد الروابط التي يتكوَّن منها المسار.

عامل مشترك أكبر: هو أكبر عدد يقبل عددان صحيحان القسمةَ عليه.

عدد أويلر: هو الثابت الرياضي e ويساوي تقريبًا ٢٫٧١٨٢٨. وهو حد $(1+1/n)^n$، حيث قيمة n تقارب اللانهاية.

عشوائية: استخدام العشوائية في الخوارزميات. بتلك الطريقة، قد تتمكَّن خوارزميةٌ ما من إيجاد حلول جيدة للمسألة في أغلب الحالات، حتى لو لم يكن إيجاد الحل الأمثل مجديًا من الناحية الحاسوبية.

عقدة: عنصر في مختلف بنى البيانات. ويُطلق على العناصر في القائمة اسم العُقَد.

عقدة متدلية: في خوارزمية بيج رانك، هي العقدة التي تحتوي على حواف واردة ولا تحتوي على حواف صادرة.

فرط الاستعداد: مصطلح يعادل التعلُّم بالصم في تعلم الآلة. يتبع النموذج الذي نحاول أن ندرِّبه بيانات التدريب بدقة شديدة لدرجةٍ تجعله متلائمًا معها إلى حد مفرط. ونتيجة لذلك، لا يتنبأ بالقيم الصحيحة للبيانات الأخرى المجهولة.

فرِّق تَسُد: طريقة لحل المسائل من خلال تقسيم المسألة إلى مسائل أصغر (مسألتين عادة)، ثم تقسيم المسائل الأصغر إلى أن نحصل على مسائل صغيرة فيصبح إيجاد الحل مباشرًا وواضحًا.

فئة التعقيد: مجموعة مسائل تتطلب المقدار نفسه من أحد الموارد (مثل الوقت أو الذاكرة) كي تُحَل.

فيض: تجاوز نطاق القيم المسموح بها على الكمبيوتر.

قانون مور: الملاحظة التي أبداها جوردون مور مؤسِّس شركتي فيرتشايلد لأشباه الموصلات وإنتل في عام ١٩٦٥، أن عدد الترانزستورات في دارة مدمجة يتضاعف كل عامين تقريبًا. وتُعَد مثالًا على النمو الأسي.

قائمة: بنية بيانات تحتوي على عناصر. يشير كل عنصر إلى العنصر التالي له باستثناء العنصر الأخير الذي لا يشير إلى شيء أو يشير إلى قيمة فارغة كما نقول. لذا تكون العناصر مرتبطة أحدها بالآخر ويطلق على مثل هذه القوائم اسم القائمة المترابطة.

قيمة فارغة: اللاشيء على الكمبيوتر.

كمبيوتر كمي: جهاز كمبيوتر يستفيد من ظواهر الكم لإجراء عمليات حاسوبية. تعمل أجهزة الكمبيوتر الكمية بوحدات الكيوبت بدلًا من وحدات البت. كذلك يمكن حل بعض المسائل على أجهزة الكمبيوتر الكمية على نحو أسرع بكثير من الأجهزة العادية. ينطوي تصنيع أجهزة الكمبيوتر الكمية على تحديات مادية صعبة.

كيوبت: الوحدة الأساسية للمعلومات الكمية. يمكن أن توجد وحدة الكيوبت في تراكب من حالتين وهما القيمة ٠ والقيمة ١ حتى نقيسها، وحينها تتدرج تحت واحدة من القيمتين الثنائيتين. يمكن تنفيذ وحدات الكيوبت باستخدام خصائص الكم مثل دوران الإلكترون.

لغة البرمجة: لغة اصطناعية يمكن استخدامها لوصف الخطوات الحاسوبية. يمكن تنفيذ لغة البرمجة على جهاز كمبيوتر. ومثل لغة البشر، لغة البرمجة لها تراكيب وقواعد لغوية تحدِّد ما يمكن كتابته فيها. يوجد العديد من لغات البرمجة، ولا يتوقَّف تطوير لغات برمجة جديدة سعيًا إلى رفع الاستفادة من البرمجة (أو لأن العديد من المبرمجين لا يستطيعون مقاومة الرغبة في إنشاء لغةٍ خاصة بهم ويأمُلون أن تُستخدم على نطاق واسع). يمكن أن تكون لغة البرمجة ذات مستوًى عالٍ عندما تبدو مشابهة إلى حدٍّ ما للغة الإنسان، أو ذات مستوًى منخفض عندما تكون عناصرها الأساسية بدائية ما يعكس مكوِّنات الجهاز الأساسية.

لوغاريتم: معكوس الرفع إلى الأُس. اللوغاريتم هو الإجابة على السؤال: «إلى أي قوة أُسية ينبغي أن أرفع العدد كي أحصل على القيمة التي أريدها؟» فإذا سألنا: «إلى أي قوة أسية ينبغي أن أرفع العدد ١٠ للحصول على العدد ١٠٠٠؟» فستكون الإجابة هي ٣ لأن ١٠^٣ = ١٠٠٠. العدد الذي نرفعه إلى الأُس يسمَّى أساس اللوغاريتم. نكتب الصيغة $log_a x = b$ إذا كانت $a^x = b$. أما إذا كانت $a = 2$، نكتب lgx.

متَّجه: صفٌّ أفقي أو عمود رأسي من الأعداد (أو المقادير الرياضية بوجه أعم). عادةً ما نجد المتجهات في علم الهندسة حيث تكون شكلًا هندسيًّا له طول واتجاه، ويعبَّر عنه بصفٍّ أو عمود يحتوي على الإحداثيات العددية؛ ولكن فكرة المتجه أعمُّ من ذلك، ومنها على سبيل المثال متجه ترتيب الصفحات. ويُعَد المتجه حالة خاصة من المصفوفة.

متَّجه ترتيب الصفحات: متَّجه يحتوي على ترتيبات الصفحات في تمثيل بياني.

متجهٌ ذاتيٌّ: في الجبر الخطي، المتَّجه الذاتي هو متجه، عند ضربه في مصفوفة معينة، تكون النتيجة هي المتجه نفسه مضروبًا في رقم ما؛ وهذا الرقم هو قيمته الذاتية. توجد خوارزمية بيج رانك المتجه الذاتي الأول لمصفوفة جوجل؛ أي المتجه الذاتي لمصفوفة جوجل ذات القيمة الذاتية الأكبر التي تساوي واحدًا.

متصفح عشوائي: شخص يتصفَّح شبكة الإنترنت بالتنقل من صفحة إلى أخرى، ويختار الصفحة التالية طبقًا للاحتمالية المعطاة من مصفوفة جوجل.

مجموعة بيانات الاختبار: بيانات نضعها جانبًا في أثناء التدريب بحيث يمكننا استخدامها للتحقُّق من مدى جودة أداء طريقة معينة في تعلُّم الآلة عندما تتعامل مع بيانات من العالم الواقعي.

مجموعة بيانات التدريب: بيانات نستخدمها مع خوارزميات تعلم الآلة لتدريبها على حل المسائل.

مجموعة متعدِّدة: مجموعة يمكن أن يظهر فيها العنصر عدة مرات؛ في الرياضيات، لا يمكن أن يظهر عنصر في مجموعة عادية أكثر من مرة.

المُدخلات الخاطئة تعطي مُخرجات خاطئة: إذا غذَّينا البرنامج بمُدخلات خاطئة بدلًا من الصحيحة، فلا ينبغي أن نتوقَّع منه المعجزات؛ سيؤتينا البرنامج مخرجاتٍ خاطئة بدلًا من المُخرجات الصحيحة التي نتوقَّعها.

مُدخلات موزونة (الخلية العصبية): مجموع حواصل ضرب المُدخلات في أوزان الخلية العصبية.

مرحلة: المرحلة في تعلُّم الآلة هي اجتياز كامل مجموعة بيانات التدريب في أثناء التدريب.

مساحة البحث: نطاق القيم الذي نبحث فيه.

مسار: تسلسل الحواف الذي يربط سلسلةً متعاقبة من العُقَد في التمثيل البياني.

مسار التنفيذ: سلسلة من الخطوات التي تنفِّذها الخوارزمية في أثناء تطبيقها.

مسار أويلري: مسار يمر عبر التمثيل البياني بحيث لا يمر على كل حافة أكثرَ من مرة. يطلق عليه أيضًا الطريق الأويلري.

مسألة البائع المتجول: مسألة تدور حول إذا ما كانت لديك قائمة بعدة مدن والمسافات بين كل مدينتين فيها، فما أقصر مسار محتمل يمكن لشخصٍ أن يتخذه لزيارة كل مدينة مرة واحدة ثم يعود إلى المدينة الأصلية؟ ربما تكون تلك المسألة هي أشهر المسائل المستعصية على الحل.

مسألة التوقُّف الأمثل: مسألة معرفة أفضل وقتٍ للتوقُّف عند محاولة تعظيم مكافأة أو تقليل عقوبة.

مسألة السكرتيرة: إحدى مسائل التوقُّف الأمثل. من بين مجموعة من المرشحين، ندرُس كل مرشح تِباعًا. يجب اتخاذ القرار بالتوظيف من عدمه في حينه، من دون أن يكون لدينا القدرة على الرجوع في قرارات سبق اتخاذها ومن دون دراسة المرشحين المتبقين.

مسألة تقليل القيمة: مسألة نحاول فيها إيجاد الحل ذي القيمة الأقل من بين الحلول المحتملة.

مسألة مستعصية على الحل: مسألة تستغرق فيها أفضل الخوارزميات التي نعرفها وقتًا طويلًا للغاية كي تحل أي شيء فيها عدا الحالات التافهة.

مستوًى فائق: تعميم المستوى في أكثر من ثلاثة أبعاد.

مشتقة: منحنى الدالة عند نقطةٍ ما؛ ويساويها معدَّل التغيير في الدالة. على سبيل المثال، التسارع هو مشتقة السرعة (معدَّل تغيُّر السرعة بالنسبة إلى الزمن).

مشتقة جزئية: مشتقة الدالة بالنسبة إلى أحد المتغيرات في الدالة ذات المتغيرات المتعددة بحيث تجعل كل المتغيرات الأخرى ثابتة.

مصفوفة: نسق مستطيل يتكوَّن عادة من الأعداد أو المقادير الرياضية بوجهٍ أعمَّ. تُرتَّب محتويات المصفوفة أفقيًّا في صفوف ورأسيًّا في أعمدة.

مصفوفة الارتباطات التشعُّبية: مصفوفة تعبِّر عن بنية تمثيل بياني؛ وهي تشبه مصفوفة التجاور ولكن مع تقسيم عناصر الصف على عدد العناصر غير الصفرية في الصف.

مصفوفة الأصفار: مصفوفة تساوي معظم عناصرها صفرًا.

مصفوفة التجاور: مصفوفة تعبِّر عن تمثيل بياني. تحتوي هذه المصفوفة على صفٍّ وعمود لكل رأس في التمثيل البياني. محتوياتها هي العدد ١ للتعبير عن كل إدخال

يتطابق صفُّه وعموده مع رأسين متصلين بحافة في التمثيل البياني؛ أما جميع المُدخلات الأخرى فتساوي صفرًا.

مصفوفة جوجل: مصفوفة من نوع خاص (نسخة معدَّلة من مصفوفة الارتباطات التشعُّبية) تُستخدم في طريقة الأس في خوارزمية بيج رانك.

المصنِّف: برنامج يصنِّف ملاحظةً ما ضمن واحدة من عدد من الفئات المحتملة.

مضروب: مضروب العدد الطبيعي n هو حاصل ضرب جميع الأعداد من ١ وحتى العدد n وفيهم العدد n نفسه. نستخدم الرمز $n!$ وبذلك تصبح الصيغة $n! = 1 \times 2 \times \cdots \times n$. ويمكن أن يمتد التعريف ليشمل كل الأعداد الحقيقية، ولكن هذا لا يعنينا هنا.

مفتاح: جزء من سجلٍّ نستخدمه لترتيبه أو البحث عنه. يمكن أن يكون المفتاح مفردًا عندما لا يمكن تفكيكه إلى أجزاء أصغر (مثل رقم الهوية) أو مركَّبًا عندما يتكوَّن من أجزاء أصغر من البيانات (مثل الاسم الكامل الذي يتكوَّن من الاسم الأول والاسم الأوسط واللقب).

مكوِّنات الأجهزة: المكوِّنات المادية التي يتكوَّن منها جهاز كمبيوتر أو جهاز رقمي. وهو مصطلح مكمِّل لمصطلح برمجيات.

ملاءمة: هي عملية التعلم من البيانات في تعلُّم الآلة. في هذه العملية، نبني نموذجًا يتلاءم مع الملاحظات.

مؤشر: مكان في ذاكرة الكمبيوتر يحمل عنوان مكان آخر في ذاكرة الكمبيوتر. وبذلك، فإن المؤشر الأول يشير إلى التالي.

مؤشر الألوان: في تلوين التمثيل البياني، هو أقل عدد ألوان مطلوب لتلوين الحواف في التمثيل البياني.

نص تشعُّبي: نص يحتوي على ارتباطات تشعُّبية.

نظام العد الأحادي: نظام أعداد يستخدم رمزًا واحدًا للتعبير عن الأعداد؛ على سبيل المثال، تعبِّر الضغطة عن وحدة، ولذا فإن III تعبِّر عن ثلاث وحدات.

النقل إلى المقدمة: خوارزمية بحث ذاتية التنظيم. عندما نجد العنصر الذي نبحث عنه، ننقله إلى الموضع الأول.

نمو أُسي: نمط نمو يُضرب فيه عدد العناصر في نفسه تِباعًا. على سبيل المثال، قد نبدأ بالعدد a من العناصر، ثم سنحصل على $a \times a$ من العناصر، وبعد ذلك $a \times a \times a$، وتصبح الصيغة بوجه عام $\overbrace{a \times a \times \cdots \times a}^{n} = a^n$. تنمو الأعداد بسرعة باستخدام النمو الأُسي.

وحدة المعالجة المركزية: الرقاقة التي تنفِّذ تعليمات برنامجٍ ما على جهاز الكمبيوتر.

وحدة خطية مصححة: خلية عصبية تستخدم دالة مصحح كدالة تنشيط لها. اختصار الوحدة الخطية المصححة هو ريلو.

وحدة معالجة الرسومات: رقاقة مصمَّمة خصيصى للتعامل مع تعليمات إنشاء الصور على الكمبيوتر ومعالجتها.

وزن (التمثيل البياني): عدد مرفَق بإحدى حواف التمثيل البياني. قد يمثِّل هذا الرقم، على سبيل المثال، مكافأة أو عقابًا مرتبطين بالرابط بين العُقَد المتصلة بواسطة الحافة.

وزن (الخلية العصبية): قيمة عددية مرتبطة بتشابك عصبي في خلية عصبية. من كل تشابك عصبي، تتلقى الخلية العصبية إدخالًا مضروبًا في وزن التشابك العصبي.

وقت خطي: نسبة الوقت إلى مُدخلات الخوارزمية، وتُكتب بالصيغة $O\,(n)$.

وقت لوغاريتمي: نسبة الوقت إلى لوغاريتم مُدخلات الخوارزمية، مثل $O\,(lgn)$. تستغرق خوارزميات البحث الجيدة وقتًا لوغاريتميًّا.

وقت لوغاريتمي خطي: نسبة الوقت إلى حاصل ضرب حجم المُدخلات ولوغاريتم مُدخلات الخوارزمية، مثل $O\,(nlgn)$. تستغرق خوارزميات الفرز الجيدة وقتًا خطيًّا لوغاريتميًّا.

وقت متعدد الحدود: نسبة الوقت إلى مُدخلات الخوارزمية مرفوعة إلى أسٍّ ثابت مثل $O\,(n^2)$.

ملاحظات

مقدمة

(1) For these and more indicators of the global progress achieved through the ideas of the Enlightenment, see Pinker 2018.

الفصل الأول: ما هي الخوارزمية؟

(1) “The Algorithmic Age” was aired on February 8, 2018, on *Radio Open Source*.

(2) For an account of algorithms in ancient Babylon, see Knuth 1972.

(3) The algorithm for distributing a number of pulses in timing slots in the SNS was given by Eric Bjorklund (1999). Godfried Toussaint (2005) noticed the parallel with rhythms, and his work is the basis for our exposition. For a more extensive discussion, see Demaine et al. 2009. For a book-length treatment of algorithms and music, see Toussaint 2013.

(4) The criteria come from Donald Knuth (1997, sec. 1), who also starts his exposition with Euclid’s algorithm.

(5) For a discussion of the enumeration of the paths on the grid, see Knuth 2011, 253–255; it is the source for the example and path images.

For the algorithm that gives the number of possible paths, see Iwashita et al. 2013.

(6) For these number descriptions, see Tyson, Strauss, and Gott 2016, 18–20. In Dave Eggers's novel *The Circle*, a thinly disguised technology company calculates the number of grains of sand in the Sahara Desert.

(7) To fold paper n times, the paper must be large enough. If you fold it always along the same dimension, you will need a long sheet of paper. The length is given by the formula $L = \frac{\pi t}{6}(2^n + 4)(2^n - 1)$, where t is the paper's thickness and n is the number of folds. If you fold a square sheet of paper in alternate directions, then the width of the square must be $W \approx \pi t 2^{(3/2)(n-1)}$. The reason why the formulas are more complicated than simple powers of two is that every time you fold the paper, you lose some part of it as it curves along the edge of the fold; it's from calculating these curves that π enters the picture in these formulas. The formulas were found in 2002 by Britney Crystal Gallivan, then a junior in high school. She went on to demonstrate that a 1,200 meters–long sheet of toilet paper could be folded in half 12 times. For a nice introduction to the power of powers (including this example), see Strogatz 2012, chapter 11.

(8) "Transistor Count," Wikipedia, https://en.wikipedia.org/wiki/Transistor_count.

(9) That is because to compare n items between them, you need to take one of them and compare it to all the other $n - 1$ items, then you take another one and compare it to the other $n - 2$ items (you have already compared it to the first item you used), and so on. That gives $1 + 2 + \cdots + (n - 1) = n(n - 1)/2$ comparisons. Then you get $O(n(n - 1)/2) = O(n^2 - n/2) = O(n^2)$, because according to the definition of big O, if your algorithm runs in time $O(n^2)$, it will certainly run in time $O(n^2 - n/2)$.

الفصل الثاني: التمثيلات البيانية

(1) Image retrieved from the Wikipedia Commons at https://commons.wikimedia.org/wiki/File:Konigsberg_Bridge.png. The image is in the public.

(2) The paper (Eulerho 1736) is available from the Euler Archive (http://eulerarchive.maa.org), maintained by the Mathematical Association of America. For an English translation, see Biggs, Lloyd, and Wilson 1986.

(3) The literature on graphs is vast, as is the subject itself. For a good starting point, see Benjamin, Chartrand, and Zhang 2015.

(4) Image from the original publication (Eulerho 1736) retrieved from the Wikipedia Commons at https://commons.wikimedia.org/wiki/File:Solutio_problematis_ad_geometriam_situs_pertinentis,_Fig._1.png. The image is in the public domain.

(5) Image from Kekulé 1872, retrieved from the Wikipedia at https://en.wikipedia.org/wiki/Benzene#/media/File:Historic_Benzene_Formulae_Kekul%C3%A9_(original).png. The image is in the public domain.

(6) For the original publication in German see Hierholzer 1873.

(7) For more details on Hierholzer's algorithm and other algorithms for Eulerian paths, see Fleischner 1991. For the use of graphs in genome assembly, see Pevzner, Tang, and Waterman 2001; Compeau, Pevzner, and Tesler 2011.

(8) For an analysis of the optimality of the greedy algorithm for online edge coloring, as well as the example of the starlike graph to show the worst case, see Bar-Noy, Motwani, and Naor 1992.

(9) In the original fable, the two characters are an ant and cicada. These two characters also feature in Latin translations of the original ancient Greek and Jean de La Fontaine's retelling of the fable in French.

(10) The invention episode is recounted by Dijkstra in his interview in Misa and Frana 2010.

الفصل الثالث: البحث

(1) For the first description of the Matthew effect, see Merton 1968. For overviews of the range of phenomena manifesting unequal distributions, see Barabási and Márton 2016; West 2017. For the stadium height and wealth disparity, see Taleb 2007.

(2) John McCabe (1965) presented a self-organized search. For analyses of the performance of the move-to-front and transposition methods, see Rivest 1976; Bachrach, El-Yaniv, and Reinstädtler 2002.

(3) The secretary problem appeared in Martin Gardner's column in February 1960 in *Scientific American*. A solution was given in the March 1960 issue. For its history, see Ferguson 1989. J. Neil Bearden (2006) provided the solution for the not all-or-nothing variant. Matt Parker (2014, chapter 11) presents the problem, along with several other mathematical ideas and an introduction to computers.

(4) Binary search goes back to the dawn of the computer age (Knuth 1998). John Mauchly, one of the designers of the ENIAC, the first general-purpose electronic digital computer, described it in 1946. For the checkered history of binary search, see Bentley 2000; Pattis 1988; Bloch 2006.

الفصل الرابع: الترتيب

(1) Hollerith 1894.

(2) Selection and insertion sort have been with us since the dawn of computers; they were included in a survey of sorting published in the 1950s (Friend 1956).

(3) According to Knuth (1998, 170), the idea behind radix sort that we have seen here seems to have been around at least since the 1920s.

(4) Flipping the coin 226 times follows from $1/52! \approx (1/2)^{226}$. The example of picking an atom from the earth is from David Hand (2014), according to whom probabilities less than one in 10^{50} are negligible on the cosmic scale.

(5) See Hoare 1961a, 1961b, 1961c.

(6) For more on randomized algorithms, see Mitzenmacher and Upfal 2017.

(7) For an account of von Neumann's life and the environment around the origins of digital computers, see Dyson 2012. For a presentation of von Neumann's merge sort program, see Knuth 1970.

الفصل الخامس: خوارزمية بيج رانك

(1) The original PageRank algorithm was published by Brin and Page (1998). We glossed over the mathematics used by the algorithm. For a more in-depth treatment, see Bryan and Leise 2006. For an introduction to search engines and PageRank, see Langville and Meyer 2006; Berry and Browne 2005. Apart from PageRank, another important algorithm used for ranking is Hypertext Induced Topic Search, or HITS (Kleinberg 1998, 1999), developed before PageRank. Similar ideas had been developed in other fields (sociometry, the quantitative study of social relationships, and econometrics, the quantitative study of economic principles) much earlier, going back to the 1940s (Franceschet 2011).

الفصل السادس: التعلُّم العميق

(1) Although today we can use technology to see neurons in much greater detail, Ramón y Cajal was a pioneer, and his drawings rank among the most elegant illustrations in the history of science. You can find neuron

images aplenty on the web, but this image is enough for us, and a simple web search should convince you of the beauty and enduring power of Ramón y Cajal's illustrations. The image is in the public domain, retrieved from https://commons.wikimedia.org/wiki/File:PurkinjeCell.jpg.

(2) To be accurate, sigmoid would refer to the Greek letter sigma, which is Σ, yet its appearance is closer to the Latin S.

(3) The tangent of an angle is defined as the ratio of the opposite side to the adjacent side in a straight triangle, or equivalently, by the sine of the angle divided by the cosine of the angle in the unit circle. The hyperbolic tangent is defined as the ratio of the hyperbolic sine by the hyperbolic cosine of an angle on a hyperbola.

(4) Warren McCulloch and Walter Pitts (1943) proposed the first artificial neuron. Frank Rosenblatt (1957) described the Perceptron. If they are more than half a century old, how come neural networks have become all the rage recently? Marvin Minsky and Seymour Papert (1969) struck a major blow to Perceptrons in their famous book of the same name, which showed that a single Perceptron had fundamental computing limitations. This, coupled with the hardware limitations of the time, ushered in a so-called winter in neural computation, which lasted well until the 1980s, when researchers found how to build and train complex neural networks. Interest in the field then revived, but still a lot more work was required to advance neural networks to the media-grabbing results that we have been seeing in the last few years.

(5) One of the challenges in neural networks is that the notation can be off-putting and hence the material seems approachable only to the initiated. In fact, it is not that complicated once you know what it is about. You often see derivatives; the derivative of a function $f(x)$ with respect to x is written $\frac{df(x)}{dx}$. The partial derivative of a function f of many variables, say, $x_1, x_2, \ldots, x_n$, is written $\frac{\partial f}{\partial x_i}$. The gradient is written $\nabla f = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}\right)$.

(6) The backpropagation algorithm came onto the scene in the mid-1980s (Rumelhart, Hinton, and Williams 1986), although various derivations of it had appeared back in the 1960s.

(7) This image is from the Fashion-MNIST data (Xiao, Rasul, and Vollgraf 2017), which was developed as a benchmark data set for machine learning. This section was inspired by the basic classification Tensor Flow tutorial at https://www.tensorflow.org/tutorials/keras/basic_classification.

(8) For a description of the first system to beat the Go human champion, see Silver et al. 2016. For an improved system that does not require human knowledge in the form of previously played games, see Silver et al. 2017.

(9) The literature on deep learning is vast. For a comprehensive introduction to the topic, see Goodfellow, Bengio, and Courville 2016. For a shorter and more approachable treatment, see Charniak 2018. For a concise overview, see LeCun, Bengio, and Hinton 2015. For deep and machine learning, see Alpaydin 2016. For a survey of automated neural architecture search methods, see Elsken, Hendrik Metzen, and Hutter 2018.

الخاتمة

(1) Besides Turing, other names on the short list were Mary Anning, Paul Dirac, Rosalind Franklin, William Herschel and Caroline Herschel, Dorothy Hodgkin, Ada Lovelace and Charles Babbage, Stephen Hawking, James Clerk Maxwell, Srinivasa Ramanujan, Ernest Rutherford, and Frederick Sanger. Babbage, Lovelace, and Turing were all computer pioneers. Babbage (1791–1871) invented the first mechanical computer and developed the essential ideas of modern computers. Lovelace (1815–1852), the daughter of Lord Byron, worked with Babbage, recognized the potential

of his invention, and was the first to develop an algorithm that would run on such a machine. She is now considered to have been the first computer programmer. For the £50 design, see the official announcement at https://www.bankofengland.co.uk/news/2019/july/50-pound-banknote-character-announcement.

(2) See the excellent biography by Andrew Hodges (1983). Turing's role in breaking the German Enigma cryptographic machine were dramatized in the 2014 film *The Imitation Game*.

(3) For a description of the machine, see Turing 1937, 1938.

(4) The Turing machine example is adapted from John Hopcroft, Rajeev Motwani, and Jeffrey Ullman (2001, chapter 8). The figure is based on Sebastian Sardina's example at http://www.texample.net/tikz/examples/turing-machine-2/.

(5) For more on the Church–Turing thesis, see Lewis and Papadimitriou 1998, chapter 5. For a discussion of the history of the Church–Turing thesis and various variants, see Copeland and Shagrir 2019.

المصادر

Alpaydin, Ethem. 2016. *Machine Learning*. Cambridge, MA: MIT Press.

Bachrach, Ran, Ran El-Yaniv, and Martin Reinstädtler. 2002. "On the Competitive Theory and Practice of Online List Accessing Algorithms." *Algorithmica* 32 (2): 201–245.

Barabási, Albert-László, and Pósfai Márton. 2016. *Network Science*. Cambridge: Cambridge University Press.

Bar-Noy, Amotz, Rajeev Motwani, and Joseph Naor. 1992. "The Greedy Algorithm Is Optimal for Online Edge Coloring." *Information Processing Letters* 44 (5): 251–253.

Bearden, J. Neil. 2006. "A New Secretary Problem with Rank-Based Selection and Cardinal Payoffs." *Journal of Mathematical Psychology* 50:58–59.

Benjamin, Arthur, Gary Chartrand, and Ping Zhang. 2015. *The Fascinating World of Graph Theory*. Princeton, NJ: Princeton University Press.

Bentley, Jon. 2000. *Programming Pearls*. 2nd ed. Boston: Addison-Wesley.

Berry, Michael W., and Murray Browne. 2005. *Understanding Text Engines: Mathematical Modeling and Text Retrieval*. 2nd ed. Philadelphia: Society for Industrial and Applied Mathematics.

Biggs, Norman L., E. Keith Lloyd, and Robin J. Wilson. 1986. *Graph Theory, 1736–1936*. Oxford: Clarendon Press.

Bjorklund, Eric. 1999. "The Theory of Rep-Rate Pattern Generation in the SNS Timing System." SNS-NOTE-CNTRL-99. Spallation Neutron Source. https://ics-web.sns.ornl.gov/timing/Rep-Rate%20Tech%20Note.pdf.

Bloch, Joshua. 2006. "Extra, Extra—Read All about It: Nearly All Binary Searches and Mergesorts Are Broken." *Google AI Blog*, June 2. http://googleresearch.blogspot.it/2006/06/extra-extra-read-all-about-it-nearly.html.

Brin, Sergey, and Lawrence Page. 1998. "The Anatomy of a Large-Scale Hypertextual Web Search Engine." *Computer Networks and ISDN Systems* 30 (1–7): 107–117.

Bryan, Kurt, and Tanya Leise. 2006. "The $25,000,000,000 Eigenvector: The Linear Algebra behind Google." *SIAM Review* 48 (3): 569–581.

Charniak, Eugene. 2018. *Introduction to Deep Learning*. Cambridge, MA: MIT Press.

Compeau, Phillip E. C., Pavel A. Pevzner, and Glenn Tesler. 2011. "How to Apply de Bruijn Graphs to Genome Assembly." *Nature Biotechnology* 29 (11): 987–991.

Copeland, B. Jack, and Oron Shagrir. 2019. "The Church-Turing Thesis: Logical Limit or Breachable Barrier?" *Communications of the ACM* 62 (1): 66–74.

Demaine, Erik D., Francisco Gomez-Martin, Henk Meijer, David Rappaport, Perouz Taslakian, Godfried T. Toussaint, Terry Winograd, and David R. Wood. 2009. "The Distance Geometry of Music." *Computational Geometry: Theory and Applications* 42 (5): 429–454.

Dyson, George. 2012. *Turing's Cathedral: The Origins of the Digital Universe*. New York: Vintage Books.

Elsken, Thomas, Jan Hendrik Metzen, and Frank Hutter. 2018. "Neural Architecture Search: A Survey." ArXiv, Cornell University. August 16. http://arxiv.org/abs/1808.05377.

Eulerho, Leonhardo. 1736. "Solutio Problematis Ad Geometrian Situs Pertinentis." *Commetarii Academiae Scientiarum Imperialis Petropolitanae* 8:128–140.

Ferguson, Thomas S. 1989. "Who Solved the Secretary Problem?" *Statistical Science* 4 (3): 282–289.

Fleischner, Herbert, ed. 1991. "Chapter X Algorithms for Eulerian Trails and Cycle Decompositions, Maze Search Algorithms." In *Eulerian Graphs and Related Topics*, 50:X.1–X.34. Amsterdam: Elsevier.

Franceschet, Massimo. 2011. "PageRank: Standing on the Shoulders of Giants." *Communications of the ACM* 54 (6): 92–101.

Friend, Edward H. 1956. "Sorting on Electronic Computer Systems." *Journal of the ACM* 3 (3): 134–168.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. Cambridge, MA: MIT Press.

Hand, David J. 2014. *The Improbability Principle: Why Coincidences, Miracles, and Rare Events Happen Every Day*. New York: Farrar, Straus and Giroux.

Hawking, Stephen. 1988. *A Brief History of Time*. New York: Bantam Books.

Hierholzer, Carl. 1873. "Ueber die Möglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechung zu Umfahren." *Mathematische Annalen* 6 (1): 30–32.

Hoare, C. A. R. 1961a. "Algorithm 63: Partition." *Communications of the ACM* 4 (7): 321.

Hoare, C. A. R. 1961b. "Algorithm 64: Quicksort." *Communications of the ACM* 4 (7): 321.

Hoare, C. A. R. 1961c. “Algorithm 65: Find.” *Communications of the ACM* 4 (7): 321–322.

Hodges, Andrew. 1983. *Alan Turing: The Enigma*. New York: Simon and Schuster.

Hollerith, Herman. 1894. “The Electrical Tabulating Machine.” *Journal of the Royal Statistical Society* 57 (4): 678–689.

Hopcroft, John E., Rajeev Motwani, and Jeffrey D. Ullman. 2001. *Introduction to Automata Theory, Languages, and Computation*. 2nd ed. Boston: Addison-Wesley.

Iwashita, Hiroaki, Yoshio Nakazawa, Jun Kawahara, Takeaki Uno, and Shin-ichi Minato. 2013. “Efficient Computation of the Number of Paths in a Grid Graph with Minimal Perfect Hash Functions.” Technical Report TCS-TR- A-13-64. Division of Computer Science, Graduate School of Information Science, Technology, Hokkaido University.

Kekulé, August. 1872. “Ueber Einige Condensationsprodukte Des Aldehyds.” *Annalen der Chemie und Pharmacie* 162 (1): 77–124.

Kleinberg, Jon M. 1998. “Authoritative Sources in a Hyperlinked Environment.” In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, 668–677. Philadelphia: Society for Industrial and Applied Mathematics.

Kleinberg, Jon M. 1999. “Authoritative Sources in a Hyperlinked Environment.” *Journal of the ACM* 46 (5): 604–632.

Knuth, Donald E. 1970. “Von Neumann’s First Computer Program.” *Computing Surveys* 2 (4): 247–261.

Knuth, Donald E. 1972. “Ancient Babylonian Algorithms.” *Communications of the ACM* 15 (7): 671–677.

Knuth, Donald E. 1997. *The Art of Computer Programming, Volume 1: Fundamental Algorithms*. 3rd ed. Reading, MA: Addison-Wesley.

Knuth, Donald E. 1998. *The Art of Computer Programming, Volume 3: Sorting and Searching*. 2nd ed. Reading, MA: Addison–Wesley.

Knuth, Donald E. 2011. *The Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Part 1*. Upper Saddle River, NJ: Addison–Wesley.

Langville, Amy N., and Carl D. Meyer. 2006. *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton, NJ: Princeton University Press.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. "Deep Learning." *Nature* 521 (7553): 436–444.

Lewis, Harry R., and Christos H. Papadimitriou. 1998. *Elements of the Theory of Computation*. 2nd ed. Upper Saddle River, NJ: Prentice Hall.

McCabe, John. 1965. "On Serial Files with Relocatable Records." *Operations Research* 13 (4): 609–618.

McCulloch, Warren S., and Walter Pitts. 1943. "A Logical Calculus of the Ideas Immanent in Nervous Activity." *Bulletin of Mathematical Biophysics* 5 (4): 115–133.

Merton, Robert K. 1968. "The Matthew Effect in Science." *Science* 159 (3810): 56–63.

Minsky, Marvin, and Seymour Papert. 1969. *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA: MIT Press.

Misa, Thomas J., and Philip L. Frana. 2010. "An Interview with Edsger W. Dijkstra." *Communications of the ACM* 53 (8): 41–47.

Mitzenmacher, Michael, and Eli Upfal. 2017. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. 2nd ed. Cambridge: Cambridge University Press.

Parker, Matt. 2014. *Things to Make and Do in the Fourth Dimension: A Mathematician's Journey through Narcissistic Numbers, Optimal*

Dating Algorithms, at Least Two Kinds of Infinity, and More. London: Penguin Books.

Pattis, Richard E. 1988. “Textbook Errors in Binary Searching.” *SIGCSE Bulletin* 20 (1): 190–194.

Pevzner, Pavel A., Haixu Tang, and Michael S. Waterman. 2001. “An Eulerian Path Approach to DNA Fragment Assembly.” *Proceedings of the National Academy of Sciences* 98 (17): 9748–9753.

Pinker, Steven. 2018. *Enlightenment Now: The Case for Reason, Science, Humanism, and Progress*. New York: Viking Press.

Rivest, Ronald. 1976. “On Self-Organizing Sequential Search Heuristics.” *Communications of the ACM* 19 (2): 63–67.

Rosenblatt, Frank. 1957. “The Perceptron: A Perceiving and Recognizing Automaton.” Report 85-460-1. Cornell Aeronautical Laboratory.

Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. 1986. “Learning Representations by Back-Propagating Errors.” *Nature* 323 (6088): 533–536.

Silver, David, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, et al. 2016. “Mastering the Game of Go with Deep Neural Networks and Tree Search.” *Nature* 529 (7587): 484–489.

Silver, David, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, et al. 2017. “Mastering the Game of Go without Human Knowledge.” *Nature* 550 (7676): 354–359.

Strogatz, Steven. 2012. *The Joy of x: A Guided Tour of Math, from One to Infinity*. New York: Houghton Mifflin Harcourt.

Taleb, Nassim Nicholas. 2007. *The Black Swan: The Impact of the Highly Improbable*. New York: Random House.

Toussaint, Godfried T. 2005. “The Euclidean Algorithm Generates Traditional Musical Rhythms.” In *Renaissance Banff: Mathematics, Music,* *Art, Culture*, edited by Reza Sarhangi and Robert V. Moody, 47–56. Winfield, KS: Bridges Conference, Southwestern College.

Toussaint, Godfried T. 2013. *The Geometry of Musical Rhythm: What Makes a "Good" Rhythm Good?* Boca Raton, FL: CRC Press.

Turing, Alan M. 1937. "On Computable Numbers, with an Application to the Entscheidungsproblem." *Proceedings of the London Mathematical Society* S2–42: 230–265.

Turing, Alan M. 1938. "On Computable Numbers, with an Application to the Entscheidungsproblem. A Correction." *Proceedings of the London Mathematical Society* S2–43: 544–546.

Tyson, Neil deGrasse, Michael Abram Strauss, and Richard J. Gott. 2016. *Welcome to the Universe: An Astrophysical Tour*. Princeton, NJ: Princeton University Press.

West, Geoffrey. 2017. *Scale: The Universal Laws of Life, Growth, and Death in Organisms, Cities, and Companies*. London: Weidenfeld and Nicholson.

Xiao, Han, Kashif Rasul, and Roland Vollgraf. 2017. "Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms." August 28. https://arxiv.org/abs/1708.07747.

قراءات إضافية

Broussard, Meredith. 2018. *Artificial Unintelligence: How Computers Misunderstand the World*. Cambridge, MA: MIT Press.

Christian, Brian, and Tom Griffiths. 2016. *Algorithms to Live By: The Computer Science of Human Decisions*. New York: Henry Holt and Company.

Cormen, Thomas H. 2013. *Algorithms Unlocked*. Cambridge, MA: MIT Press.

Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. *Introduction to Algorithms*. 3rd ed. Cambridge, MA: MIT Press.

Denning, Peter J., and Matti Tedre. 2019. *Computational Thinking*. Cambridge, MA: MIT Press.

Dewdney, A. K. 1993. *The (New) Turing Omnibus: 66 Excursions in Computer Science*. New York: W. H. Freeman and Company.

Dyson, George. 2012. *Turing's Cathedral: The Origins of the Digital Universe*. New York: Vintage Books.

Erwig, Martin. 2017. *Once upon an Algorithm: How Stories Explain Computing*. Cambridge, MA: MIT Press.

Fry, Hannah. 2018. *Hello World: How to Be Human in the Age of the Machine*. London: Doubleday.

Harel, David, and Yishai Feldman. 2004. *Algorithmics: The Spirit of Computing*. 3rd ed. Harlow, UK: Addison-Wesley.

Louridas, Panos. 2017. *Real-World Algorithms: A Beginner's Guide*. Cambridge, MA: MIT Press.

MacCormick, John. 2013. *Nine Algorithms That Changed the Future: The Ingenious Ideas That Drive Today's Computers*. Princeton, NJ: Princeton University Press.

O'Neil, Cathy. 2016. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. New York: Crown Publishing Group.

Petzold, Charles. 2008. *The Annotated Turing: A Guided Tour through Alan Turing's Historic Paper on Computability and the Turing Machine*. Indianapolis: Wiley Publishing.

Sedgewick, Robert, and Kevin Wayne. 2017. *Computer Science: An Interdisciplinary Approach*. Boston: Addison-Wesley.

هنداوي