الجزء 2. مفاهيم الخوارزمية والبرامج

1) مفهوم الخوارزمية:
2) .تمثيل الخوارزمية
3) مفاهيم الثابت والمتغير
4) التراكيب الأساسية في الخوارزميات
5) شرح طريقة عمل الخوارزميات
6) التعبير عن الخوارزميات
7) تصميم الخوارزميات
8) تحليل الخوارزميات
9) أنواع الخوارزميات واستخداماتها
10) تصنيف الخوارزميات حسب طريقة التنفيذ
11) تصنيف الخوارزميات حسب نموذج التصميم
12) تصنيف الخوارزميات حسب أسلوب التحسين
13) تصنيف الخوارزميات حسب مجال الدراسة
14) تصنيف الخوارزميات حسب درجة التعقيد

1) مفهوم الخوارزمية:

الخوارزمية عبارة عن سلسلة من التعليمات، عند تنفيذها بشكل صحيح، تؤدي إلى نتيجة معينة فالخوارزمية تحتوي فقط على تعليمات مفهومة من قبل الشخص الذي سيتعين عليه تنفيذها.

عملية حل مشاكل بالكمبيوتر

لغات البرمجة تنجز البرنامج عبارة عن سلسلة من التعليمات ، بترتيب محدد جيدًا ، والتي تعدل "حالة الذاكرة".

الذاكرة : هي مساحات "أماكن" شاغرة في الفرص الصلب تسمى "المتغيرات" ، والتي تحتوي على قيم ، عادة ما تكون أرقامًا أو نصًا .

يمكننا تخيل المتغيرات كمربعات بأسماء. يقوم البرنامج بقراءة أو تعديل القيم الموجودة في هذه المربعات.

1.1. عملية حل المشكلات

عند تنفيذ برنامج حتمي ، تمر الذاكرة بالتالي عبر سلسلة محدودة من الحالات.

1 .تحليل المشكلة: إنها مسألة تحديد الأهداف. ماذا نطلب أن نفعل؟ ما هو العمل الذي يتعين القيام به؟

2 .ضع طريقة للقرار: إنها مسألة قول كيف ، دون الخوض في التفاصيل

سيتم حل المشكلة (تحديد العمليات التي سيتم تنفيذها للحصول على حل)

3 .صياغة الخوارزمية: اكتب التعليمات المكونة للخوارزمية.

4 .ترجمة الخوارزمية: ترجمة الخوارزمية إلى لغة برمجة.

5 .تنفيذ البرنامج: أدخل البيانات اللازمة ويقوم الحاسب بمعالجتها ويقدم لنا النتيجة.

2) .تمثيل الخوارزمية

هناك طريقتان لتمثيل الخوارزمية: تمثيل رسومي بواسطة مخطط انسيابي وتمثيل نصي بواسطة رمز زائف.

2.1. التمثيل البياني (المخطط الهيكلي): يتكون المخطط التنظيمي من:

رموز تمثل نوع الإجراء الذي يتعين القيام به. يتم وصف الإجراء داخل الرمز ؛

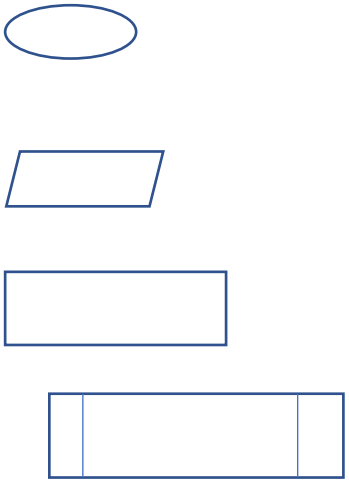
العلاقات بين هذه الرموز المختلفة ممثلة بخطوط أو أسهم. يشيرون إلى

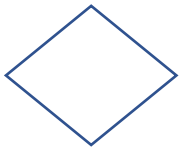
المسار الذي يجب اتباعه للوصول إلى الرمز التالي.

يبدأ المخطط الانسيابي دائمًا بمربع البدء الذي يحدد بداية الخوارزمية وينتهي بمربع النهاية الذي يشير إلى نهاية الخوارزمية.

الشكل البيضاوي: وفيه إما كلمة BEGINNING أو كلمة END ، والتي تمثل على التوالي بداية ونهاية الخوارزمية.

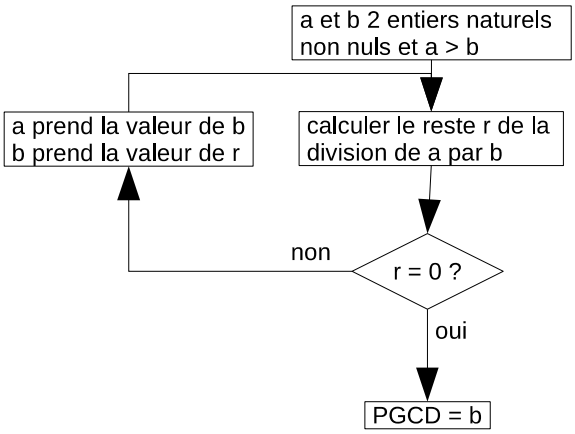
متوازي الأضلاع: وهو عبارة عن عملية تبادل بين الحاسوب والإنسان ، إما إدخال البيانات (المدخلات) ، أو عرض النتائج (المخرجات).
المستطيل: يمثل إجراءً أو معالجة بيانات أو حسابًا ؛ يمكن أن يكون هذا الإجراء بسيطًا أو معقدًا (أي مجموعة من الإجراءات).
خوارزمية فرعية (جزء من الخوارزمية يعتبر عملية بسيطة).



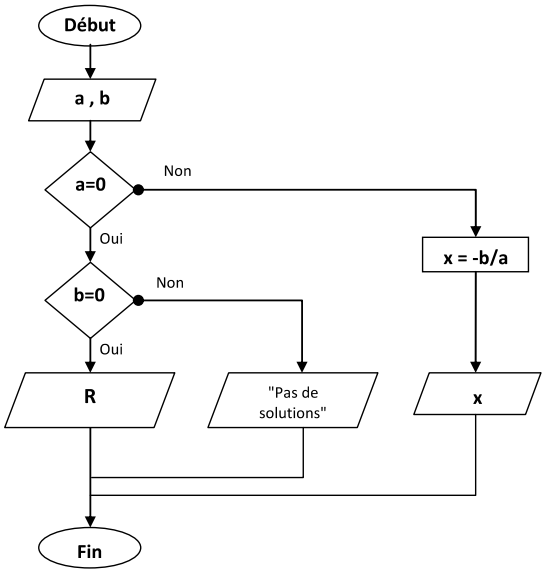


يمثل الشرط.

مثال1: مخطط خوارزمية إقليدس:



مثال 2: مخطط يحسب حل المعادلة a*x+b=0 dans R.



2.2. التمثيل النصي (رمز زائف):

الطريقة الثانية ، المستخدمة بشكل متكرر ، تستخدم لتطوير خوارزمية ، وفقًا للبنية التالية:

- ***أدخل (a، b)***
- ***a، b هي اعداد طبيعية غير منعدمة بحيث b < a***
- ***r هو باقي قسمة a علىb***
- ***مادام : r ≠0 إفعل :***
- ***a = b***
- ***b=r***
- ***r = a / b باقي قسمة***
- ***نهاية مادم***
- ***اكتب b***

الخوارزمية كالتالي:

1- كلمة خوارزمية متبوعة باسم الخوارزمية.

2-جزء التصريح: هذا الجزء مخصص للإعلان عن جميع الثوابت والمتغيرات المستخدمة في الخوارزمية.

3-جسم الخوارزمية: يحتوي هذا الجزء على مهام الخوارزمية.

3) مفاهيم الثابت والمتغير

البيانات هي قيمة يدخلها المستخدم أثناء تنفيذ البرنامج. مثال: في المثال السابق (حساب مجموع عددين A و B)، البيانات هي A و.B

النتيجة هي قيمة تنتجها الخوارزمية. يمكن أن تكون نهائية أو وسيطة. مثال:

مجموع A و B في المثال السابق هو نتيجة نهائية.

حساب Δ في معادلة من الدرجة الثانية هو نتيجة وسيطة.

المتغيرات: متغير تستخدمه خوارزمية لتخزين قيمة (بيانات أو نتيجة).

- يمكن للمتغير أن يغير القيمة أثناء الخوارزمية.
- في الواقع ، يشير المتغير إلى موقع في الذاكرة ، قد يتغير محتواه أثناء تنفيذ الخوارزمية.

الثوابت: الثابت هو قيمة ثابتة تستخدمها الخوارزمية. مثال (3.14) Pi :، و ثابت الجاذبية (9.81) ، إلخ.

المعرف: المعرف هو الاسم المستخدم لتعيين متغير أو ثابت في الخوارزمية (أي اسم متغير أو ثابت أو خوارزمية).

- يحتوي المعرف على الأرقام من 0 إلى 9 ، والحروف من A إلى Z (الأحرف الكبيرة أو الصغيرة) وشلطة 8 (_).

المعرف: أداة تعريف يجب أن تبدأ برسالة أو حزمة من الثمانية.

مثل : طريقة الصحيحة دلتا ، طول ، الحجم، س غير صحيحة a، 2Ca

النوع: يمثل نوع المتغير أو الثابت طبيعة القيم التي يمكن تخزينها في ذلك المتغير أو الثابت (الأرقام ، النص ، إلخ)

الكلمات الرئيسية: الكلمات الرئيسية هي كلمات محجوزة للاستخدام المحكم A .تعريف a مفتاح كلمة. السابق.

- **`algorithme :`** permet de définir ou de donner le nom à l'algorithme.
- **`début`** : marque le commencement de l'algorithme.
- **`fin :`** marque la fin de l'algorithme.
- **`var`**, **`const`**, **`si`**, **`sinon`**, …

4) التراكيب الأساسية في الخوارزميات

في الأصل كان يقتصر معنى الخوارزمية على تراكيب ثلاثة فقط وهي التراكيب الأساسية والتي تتمثل في كل من التسلسل والاختيار والتكرار.

لقد أُثبت أن الخوارزميات ليست بحاجة إلى تراكيب إضافية غير هذه التراكيب، فاستخدام هذه التراكيب يسهل فهم الخوارزمية واكتشاف الأخطاء الواردة فيها وتغييرها وتصحيحها. وفيما يلي وصفًا موجزًا لهذه التراكيب الثلاثة:

التسلسل: تكون الخوارزمية عبارة عن مجموعة من التعليمات المتسلسلة، وهذه التعليمات إما أن تكون بسيطة أو من النوعين الآخرين.

الاختيار: بعض المشاكل لا يمكن حلها بتسلسل بسيط للتعليمات، وقد تحتاج إلى اختبار بعض الشروط وتنظر إلى نتيجة الاختبار.

فإذا كانت النتيجة صحيحة تتبع مسار يحتوي تعليمات متسلسلة.

وإذا كانت النتيجة خاطئة تتبع مسار آخر مختلف من التعليمات.

وهذه الطريقة تسمى اتخاذ القرار أو الاختيار.

التكرار: في بعض الأحيان وعند محاولة حل بعض المشكلات قد يلزم إعادة نفس تسلسل الخطوات عدد من المرات حتى الوصول إلى الحل المطلوب. وهذا ما يُطلق عليه التكرار.

5) شرح طريقة عمل الخوارزميات

من أجل شرح وتبسيط مفهوم الخوارزمية وطريقة عملها، نفرض أنه لدينا مجموعة من الاعداد الصحيحة الموجبة التي تبدأ بالعدد (1) وتنتهي بالعدد (20).

والأعداد كما يلي:1، 2، 3، 4، 5، 6، 7، 8، 9، 10، 11، 12، 13، 14، 15، 16، 17، 18، 19، 20

ونفرض أن المطلوب معرفة العدد الذي يمثل إجابة على سؤال معين.

مثلا، لو افترضنا أنها لعبة بين شخصين، وقام الشخص الأول بتحديد عدد في ذهنه من بين هذه الأعداد، وطلب من الشخص الثاني معرفة هذا العدد بطريقة التخمين، ويقوم الشخص الأول بالرد في كل محاولة وتوضيح ما إذا كان التخمين صح أو خطأ.

وفي حالة التخمين الخطأ يحدد له موقع العدد الصحيح وما إذا كان:

- أكبر من التخمين أو
- أصغر من التخمين

الحل الاعتيادي

الأصل في هذه المسألة هو أن الشخص الثاني قد يقوم بعملية التخمين بشكل عشوائي لمعرفة العدد الذي اختاراه الشخص الأول في ذهنه، وقد يستمر في تكرار المحاولات حتى يخمن الرقم الصحيح. وأقصى عدد للمحاولات في هذه الحالة هو (20) محاولة إذا لم يحالفه الحظ في (19) محاولة وكان التخمين الصحيح هو في المحاولة الأخيرة. أو إذا بدأ بالتخمين بشكل تصاعدي متسلسل بدءًا من العدد (1) وكانت الإجابة الصحيحة هي العدد (20).

الحل بأسلوب الخوارزميات

أما في حالة تحديد موقع العدد الصحيح بعد كل تخمين، من حيث أنه أكبر من أو أصغر من التخمين المذكور، فيكون بالإمكان التوصل إلى الحل الصحيح بعد (5) محاولات على الأكثر. ويمكن إجراء هذه التجربة عمليًا للتحقق من ذلك.

إن ملخص ما يقوم به عقل الإنسان، في هذا المثال المبسط، هو تحديدًا ما تقوم به الخوارزمية في أبسط صورة لها. فعندما يقوم الشخص الثاني بالتخمين في كل مرة ويكون الرد بأن إجابته خطأ وأن العدد الصحيح هو أكبر من (أو أصغر) من التخمين، يقوم عقله بحذف التخمين مع كل الأعداد الأصغر منه (أو الأكبر منه)، والبدء من جديد في تخمين عدد من الأعداد المتبقية، وهكذا.

أي أن العقل يقوم بسلسلة من العمليات في كل مرة ويقوم بتكرار هذه العمليات بنفس النمط وبشكل متسلسل إلى أن يتوصل إلى الحل أو التخمين الصحيح في هذه اللعبة، وهذا هو ما تقوم به الخوارزميات.

3.5. الشفرة البرمجية

الشفرة البرمجية (بالإنجليزية Code) :هي شيفرة مكتوبة باستخدام لغة برمجة يفهمها الحاسوب.

وكل لغة لها قواعدها الخاصة، وتُستخدم للتعبير عن الخوارزمية.

مثلا، لكتابة شفرة خوارزمية إقليدس، وهو نفس المثال السابق والمتعلق بإيجاد القاسم المشترك الأكبر لعددين، موضحة في خطوات متسلسلة ومتداخلة عن طريق حلقات، ضمن برنامج مكتوب بلغة بيسكBasic ، وهي لغة برمجة يفهمها الحاسوب، كما يلي:

```
5 REM Euclid's algorithm for greatest common divisor

6 PRINT "Type two integers greater than 0"

10 INPUT A, B

20 IF B=0 THEN GOTO 80

30 IF A > B THEN GOTO 60

40 LET B=B-A

50 GOTO 20

60 LET A=A-B

70 GOTO 20

80 PRINT A

90 END
```

6) تصميم الخوارزميات

يُشير تصميم الخوارزميات إلى طريقة أو عملية رياضية لحل المشكلات.

يُعد تصميم الخوارزميات جزءًا من العديد من نظريات الحلول لأبحاث التشغيل، مثل البرمجة الديناميكية.

تُسمى تقنيات تصميم وتنفيذ تصميمات الخوارزمية أيضًا أنماط تصميم الخوارزمية، مثل نمط القالب ونمط الديكور.

يكمن أحد أهم جوانب تصميم الخوارزمية في إنشاء خوارزمية تتمتع بوقت تشغيل فعّال.

1.5. خطوات نموذجية في تصميم الخوارزميات

يمكن تلخيص الخطوات النموذجية التي يتم اتباعها من أجل تصميم الخوارزميات كما يلي:

1. تعريف المشكلة
2. تطوير النموذج
3. تحديد المواصفات
4. كتابة الخوارزمية
5. التحقق من صحة الخوارزمية
6. تحليل الخوارزمية
7. تنفيذ الخوارزمية
8. اختبار الخوارزمية
9. إعداد التوثيق

إن الغرض من معظم الخوارزميات هو تنفيذ برامج الكمبيوتر. ومع ذلك، يتم تنفيذ الخوارزميات أيضًا بوسائل أخرى، كما هو الحال في الشبكة العصبية البيولوجية، (على سبيل المثال، الدماغ البشري الذي ينفذ العمليات الحسابية أو حتى الحشرة عندما تقوم بالبحث عن الطعام)، أو في دائرة كهربائية، أو في جهاز ميكانيكي.

2.5. خوارزميات الكمبيوتر

في أنظمة الكمبيوتر، تُعد الخوارزمية في الأساس مثالًا للمنطق المكتوب في البرنامج بواسطة مطوري البرامج لتكون فعّالة بالنسبة لأجهزة الكمبيوتر "المستهدفة" المقصودة لإنتاج مخرجات من مدخلات معينة (أو ربما بدون مدخلات). الخوارزمية المثالية، حتى تلك التي تعمل في الأجهزة القديمة، ستُنتج نتائج أسرع من

خوارزمية غير مثالية (تستغرق وقت أكبر وفيها تعقيد أعلى) لنفس الغرض حتى وإن كانت تعمل في أجهزة أكثر كفاءة؛ وهذا هو السبب في اعتبار الخوارزميات تقنية من التقنيات مثل أجهزة الكمبيوتر.

برامج "أنيقة" (مضغوطة)، برامج "جيدة" (سريعة): تظهر فكرة "البساطة والأناقة" بشكل غير رسمي في أوساط البرمجة لوصف الخوارزمية الجيدة. أحد معايير الجودة في الخوارزمية هو قِصر الوقت المستغرق في أداءها.

المعايير الأخرى هي قدرة الخوارزمية على التكيف مع أجهزة الكمبيوتر.

والبساطة والأناقة في الخوارزمية تعني أنها أصغر برنامج ممكن لإنتاج الناتج الذي تقوم به.

لسوء الحظ، قد يكون هناك مفاضلة بين الجودة (السرعة) والأناقة (الضغط أو الاختصار)، قد يستغرق برنامج أنيق خطوات أكثر لإكمال حسابات معينة من برنامج أقل أناقة. وسوف يتم إظهار هذا الفرق في البند التالي بعنوان تحليل الخوارزميات.

ولكن ماذا عن محاكاة أو تنفيذ الشيء الحقيقي؟ يجب على المبرمج ترجمة الخوارزمية إلى لغة يمكن للمحاكي/ الكمبيوتر تنفيذها بشكل فعّال. مثال على ذلك: عند حساب جذور المعادلة التربيعية، يجب أن يعرف الكمبيوتر كيفية أخذ الجذر التربيعي، إذا لم يكن كذلك، فإن الخوارزمية، حتى تكون فعالة، يجب أن توفر مجموعة من القواعد لاستخراج الجذر التربيعي. هذا يعني أن المبرمج يجب أن يعرف "لغة" فعّالة بالنسبة لعامل الحوسبة المستهدفة بالكمبيوتر.

1.2.5.خوارزمية إيجاد العدد الأكبر

تتمثل إحدى أبسط الخوارزميات في العثور على أكبر عدد في قائمة من الأعداد المرتبة عشوائيًا.

يتطلب إيجاد الحل النظر إلى كل عدد في القائمة. ومن أجل ذلك يتم تصميم خوارزمية بسيطة، يمكن وصفها باستخدام لغة عالية المستوى بالنثر باللغة الإنجليزية على النحو التالي:

يمكن وصف هذه الخوارزمية باستخدام لغة رفيعة المستوى، وهي لغة أقرب إلى اللغة الطبيعية، كما يلي:

If there are no numbers in the set then there is no highest number.

Assume the first number in the set is the largest number in the set.

For each remaining number in the set: if this number is larger than the current largest number, consider this number to be the largest number in the set.

When there are no numbers left in the set to iterate over, consider the current largest number to be the largest number of the set.

ترجمة الوصف:

1. إذا لم تكن هناك أعداد في المجموعة، فلن يكون هناك أعلى رقم.
2. افترض أن العدد الأول في المجموعة هو أكبر عدد في المجموعة.
3. لكل عدد متبقٍ في المجموعة: إذا كان هذا العدد أكبر من أكبر عدد حالي، فضع في اعتبارك أن هذا العدد هو أكبر عدد في المجموعة.
4. عند عدم تبقي أعداد في المجموعة للتكرار، ضع في اعتبارك أن أكبر عدد حالي هو أكبر عدد في المجموعة.

2.5.2. الوصف شبه الرسمي للخوارزمية

الوصف شبه الرسمي للخوارزمية هو وصف مكتوب بالنثر ولكنه أقرب كثيرًا إلى اللغة عالية المستوى لبرنامج الكمبيوتر.

فيما يلي الترميز الأكثر رسمية للخوارزمية في الشفرة الزائفة أو شفرة الجسر:

```
Algorithm LargestNumber

  Input: A list of numbers L.

  Output: The largest number in the list L.

  if L.size = 0 return null

  largest ← L[0]

  for each item in L, do

    if item > largest, then

      largest ← item

  return largest
```

يشير الرمز "←" إلى عملية التعيين.

على سبيل المثال، "largest ← item"تعني تغيير قيمة largest وإعطاءها قيمة.item

" return "تُنهي الخوارزمية وتُعطي القيمة النهائية.

7) تحليل الخوارزميات

غالبًا ما يكون من المهم معرفة مقدار الموارد المعينة، مثل الوقت والذاكرة ومساحة التخزين، نظريًا لخوارزمية من أجل تقدير كفاءتها.

تم تطوير طرق لتحليل الخوارزميات للحصول على هذه الإجابات الكمية (أو التقديرات الكمية).

على سبيل المثال، خوارزمية الفرز وإيجاد العدد الأكبر أعلاه لها متطلب زمني يتوقف على عدد الأعداد المُدخلة والتي نريد معرفة أكبر عدد فيها. وبالنسبة لمتطلب الذاكرة ومساحة التخزين فيه تحتاج فقط إلى تذكر قيمتين خلال تنفيذها، أكبر عدد تم العثور عليه حتى الآن، وموقعه الحالي في قائمة الإدخال. لذلك، يُقال إن هذه الخوارزمية لديها متطلبات مساحة دُنيا1O ، إذا لم يتم حساب المساحة المطلوبة لتخزين أرقام الإدخال، أو مساحة عظمى On إذا تم حسابها.

قد تُكمل خوارزميات المختلفة نفس المهمة بمجموعة مختلفة من التعليمات في وقت أو مساحة أو "جهد" أقل أو أكثر من غيرها. على سبيل المثال، تتفوق خوارزمية البحث الثنائي ذات التكلفة(O-log n) على البحث المتسلسل ذات التكلفة(On) عند استخدامها لعمليات بحث في القوائم في الجدول أو في المصفوفات التي يتم فرزها.

وتفسير ذلك في المثال التالي:

خوارزمية تقوم بنفس العمل والمهمة وهي إيجاد أكبر عدد من بين مجموعة من الأعداد، ولكن بطريقة تقسيم الأعداد إلى مجموعتين متقابلتين ومقارنة كل عددين متناظرين معًا، ويتم الاحتفاظ بالأعداد الكبيرة من كلا المجموعتين في مجموعة جديدة، ثم تقسيمها إلى مجموعتين جديدتين وتكرار نفس العملية حتى الحصول بالنهاية على عددين فقط يتم المقارنة بينهما لمعرفة العدد الأكبر وإعطاء النتيجة.

1.7. التحليل التجريدي والتحليل التجريبي للخوارزميات:

يُعد تحليل ودراسة الخوارزميات أحد مسارات بحوث علوم الكمبيوتر، وغالبًا ما يُمارس بشكل تجريدي دون استخدام لغة برمجة معينة أو تنفيذ. وبهذا المعنى، يُشبه تحليل الخوارزمية التخصصات الرياضية الأخرى من حيث أنه يُركّز على الخصائص الأساسية للخوارزمية وليس على تفاصيل أي تطبيق معين.

وعادة ما يتم استخدام الشفرة الكاذبة من أجل تحليل الخوارزميات لأنه أبسط تمثيل عام. ومع ذلك، في نهاية المطاف، غالبًا ما يتم تنفيذ معظم الخوارزميات على أنظمة أجهزة أو برامج معينة ويتم اختبار كفاءة الخوارزمية في النهاية باستخدام لغة برمجة أو شفرة حقيقية. لحل مشكلة "لمرة واحدة"، قد لا يكون لكفاءة

خوارزمية معينة عواقب كبيرة (ما لم تكن n كبيرة جدًا)، ولكن بالنسبة للخوارزميات المصممة للاستخدام العلمي التفاعلي أو التجاري أو طويل المدى، فقد تكون الكفاءة أكثر أهمية. وكثيرًا ما يكشف تضخم حجم المعطيات أو المدخلات (n) خوارزميات غير فعّالة تكون جيدة في حالة المدخلات القليلة نسبيًا.

الاختبار التجريبي مفيد لأنه قد يكشف عن تفاعلات غير متوقعة تؤثر على الأداء.

يمكن استخدام المقاييس للمقارنة ما بين قبل وبعد التحسينات المحتملة لخوارزمية بعد تحسين البرنامج. لا يمكن للاختبارات التجريبية أن تحل محل التحليل التجريدي، ولكنها ضرورية للحكم بشكل عادل على كفاءة الخوارزمية.

2.7. كفاءة تنفيذ الخوارزمية

لتوضيح التحسينات المحتملة الممكنة حتى في الخوارزميات الراسخة، مثلا، يمكن أن يؤدي ابتكار هام حديث يتعلق بخوارزميات FFT المستخدمة بكثرة في مجال معالجة الصور- (*"تحويل فوريي السريع Fast Fourier Transformation هي خوارزمية تمكن من حساب قيمة تحويل فوري المنقطع بسرعة. تعود سرعة هذه الخوارزمية إلى أنها لا تقوم بحساب الأجزاء التي يساوي مجموعها صفرا في تحويل فوريي المتقطع"*).

إلى تقليل وقت المعالجة حتى 1000 مرة لتطبيقات مثل التصوير الطبي. بشكل عام، تعتمد تحسينات السرعة على الخصائص المرتبطة بالمشكلة، وهي شائعة جدًا في التطبيقات العملية. يمكّن التسريع بهذا الحجم أجهزة الحوسبة التي تُستخدم على نطاق واسع معالجة الصور (مثل الكاميرات الرقمية والمعدات الطبية) من استهلاك طاقة أقل.

8) أنواع الخوارزميات واستخداماتها

كل مجال من مجالات العلوم لديه مشاكله الخاصة ويحتاج إلى خوارزميات فعّالة من أجل حل تلك المشاكل.

كما أنه غالبًا ما تتم دراسة المشكلات ذات الصلة في مجال واحد معًا.

ومن أشهر أمثلة الخوارزميات:

- خوارزميات البحث
- خوارزميات الفرز
- خوارزميات الترتيب
- خوارزميات الدمج
- الخوارزميات الرقمية
- خوارزميات الرسم البياني

- خوارزميات السلسلة
- الخوارزميات الهندسية الحاسوبية
- الخوارزميات التوافقية
- الخوارزميات الطبية
- خوارزميات الذكاء الاصطناعي
- خوارزميات التشفير
- خوارزميات ضغط البيانات
- خوارزميات التحليل الإحصائي
- خوارزميات تنقيب البيانات أو التنقيب في قواعد البيانات

وتتداخل الحقول وميادين العلوم المختلفة مع بعضها البعض، وقد تؤدي تحسينات الخوارزمية في أحد الحقول إلى تحسين الحقول الأخرى، أو حتى الحقول غير المرتبطة بها أحيانًا. فعلى سبيل المثال، تم اختراع خوارزميات البرمجة الديناميكية لتحسين استهلاك الموارد في الصناعة، ولكنها تُستخدم الآن في حل مجموعة واسعة من المشكلات في العديد من المجالات الأخرى.

9) تصنيف الخوارزميات حسب طريقة التنفيذ

وهناك طرق مختلفة لتصنيف الخوارزميات، لكل منها مزاياها الخاصة.

فيما يلي أهم هذه التصنيفات :

1.9. الخوارزمية التكرارية

الخوارزمية التكرارية بالإنجليزية: (Recursion) هي تلك التي تستدعي (تُشير إلى) نفسها بشكل متكرر حتى يتطابق شرط معين (يُعرف أيضًا بحالة الإنهاء)، وهي طريقة شائعة للبرمجة الوظيفية.

تستخدم الخوارزمية التكرارية تركيبات متكررة مثل الحلقات بالإنجليزية (Loops) وأحيانًا هياكل بيانات إضافية مثل Stacks لحل المشكلات المُعطاة.

تتناسب بعض المشاكل بشكل طبيعي مع تنفيذ واحد أو آخر.

كل نسخة تكرارية لها نسخة تكرارية مكافئة (ولكن ربما أكثر أو أقل تعقيدًا)، والعكس بالعكس.

2.9. الخوارزمية المنطقية

يمكن اعتبار الخوارزمية على أنها خصم منطقي خاضع للرقابة. يمكن التعبير عن هذه الفكرة على النحو التالي: **خوارزمية = تحكم + منطق**

يُعبر المكون المنطقي عن البديهيات التي يمكن استخدامها في الحساب ويحدد عنصر التحكم الطريقة التي يتم بها تطبيق الخصم على البديهيات. هذا هو الأساس لنموذج البرمجة المنطقية .في لغات البرمجة المنطقية الخالصة، يكون عنصر التحكم ثابتًا ويتم تحديد الخوارزميات من خلال توفير مكون المنطق فقط. إن جاذبية هذا النهج هي دلالات أنيقة: تغيير في البديهيات ينتج عنه تغيير واضح المعالم في الخوارزمية.

3.9. الخوارزميات التسلسلية والمتوازية والموزعة

عادة ما تتم مناقشة الخوارزميات مع افتراض أن أجهزة الكمبيوتر تنفذ تعليمات واحدة من الخوارزمية في كل مرة. تسمى أجهزة الكمبيوتر هذه أحيانًا أجهزة الكمبيوتر التسلسلية. تسمى الخوارزمية المصممة لمثل هذه البيئة خوارزمية تسلسلية، على عكس الخوارزمية المتوازية أو الخوارزمية الموزعة. تستفيد الخوارزمية المتوازية من مكونات الكمبيوتر حيث يمكن للعديد من المعالجات العمل على مشكلة في نفس الوقت، بينما تستخدم الخوارزمية الموزعة أجهزة متعددة متصلة بشبكة كمبيوتر.

تُقسم الخوارزمية المتوازية أو الموزعة المشكلة إلى مشاكل فرعية أكثر تناظرية أو غير متماثلة وتجميع النتائج مرة أخرى معًا.

استهلاك الموارد في مثل هذه الخوارزميات ليس فقط في كل معالج ولكن أيضًا الاتصالات العامة بين المعالجات.

يمكن موازاة بعض خوارزميات الفرز بكفاءة، ولكن تكلفة الاتصال الخاصة بهم مكلفة.

والخوارزميات التكرارية متوازية بشكل عام.

لا تحتوي بعض المشكلات على خوارزميات متوازية وتسمى في هذه الحالة مشكلات تسلسلية أصلية أو خالصة.

4.9. الخوارزميات الحتمية وغير الحتمية

تحل الخوارزميات الحتمية المشكلة بقرار دقيق في كل خطوة من الخوارزمية في حين تحل الخوارزميات غير الحتمية المشكلات عن طريق التخمين على الرغم من أن التخمينات النموذجية يتم جعلها أكثر دقة من خلال استخدام الاستدلال.

5.9. الخوارزميات الدقيقة أو التقريبية

بينما تصل العديد من الخوارزميات إلى حل دقيق، تبحث خوارزميات التقريب عن تقريب أقرب إلى الحل الحقيقي.

يمكن الوصول إلى التقريب إما باستخدام استراتيجية حتمية أو عشوائية.

هذه الخوارزميات لها قيمة عملية للعديد من المشاكل الصعبة. أحد أمثلة الخوارزمية التقريبية هي مشكلة الحقيبة، حيث توجد مجموعة من العناصر المعطاة هدفها هو حزم الحقيبة للحصول على أقصى قيمة إجمالية. كل عنصر له بعض الوزن وبعض القيمة. الوزن الإجمالي الذي يمكن حمله لا يزيد عن بعض الرقم الثابت Xلذا، يجب أن يأخذ الحل في الاعتبار أوزان العناصر بالإضافة إلى قيمتها.

5.9. الخوارزميات الكمية

تعمل الخوارزمية الكمية على نموذج واقعي للحساب الكمي.

يستخدم المصطلح عادة لتلك الخوارزميات التي تبدو كميّة بطبيعتها، أو تستخدم بعض السمات الأساسية للحوسبة الكمية مثل التراكب الكمي أو التشابك الكمي.

10) تصنيف الخوارزميات حسب نموذج التصميم

طريقة أخرى لتصنيف الخوارزميات هي بحسب نموذج التصميم أو المنهيجة.

هناك عدد معين من النماذج، يختلف كل منها عن الآخر.

علاوة على ذلك، تتضمن كل فئة من هذه الفئات العديد من أنواع الخوارزميات المختلفة.

بعض النماذج الشائعة هي كما يلي:

1.10. خوارزمية البحث الشامل

خوارزمية البحث الشامل او المعروفة باسم خوارزمية التوليد والاختبار أو القوة الغاشمة، هي خوارزمية عامة جدًا لحل المشكلات ونموذج خوارزمي يتكون من تعداد جميع الإمكانات المحتلمة بشكل منهجي والتحقق مما إذا كان كل احتمال يشكل حلا للمشكلة.

وتُعتير هذه الطريقة الساذجة في الخوارزميات لتجربة كل حل ممكن لمعرفة أيهما أفضل، ويمكن تسميتها الخوارزميات الساذجة.

2.10. خوارزميات فرق تسدد

تقوم خوارزمية فرق تسد (بالإنجليزية Divide and Conquer) :بشكل متكرر بتقليص المشكلة إلى مشكلة واحدة أو أكثر من المشكلات المثيلة للمشكلة الأصلية ولكنها أصغر منها، وتقوم بذلك عادة بشكل متكرر حتى تكون المشكلات صغيرة بما يكفي لحلها بسهولة. أحد الأمثلة على خوارزمية فرق تسد خوارزمية دمج الفرز. يمكن إجراء الفرز على كل جزء من البيانات

بعد تقسيم البيانات إلى شرائح ويمكن الحصول على فرز البيانات بالكامل عن طريق دمج الأجزاء. نموذج مبسط من خوارزمية فرق تسد هو خوارزمية تُسمى اختزل تسد، والتي تحل مشكلة فرعية متطابقة مع المشكلة الأصلية وتستخدم حل هذه المشكلة الفرعية لحل المشكلة الأكبر.

خوارزميات فرق تسد تُقسم المشكلة إلى مشاكل فرعية متعددة، وبالتالي فإن الوصول إلى الحل فيها هو أكثر تعقيدًا من خوارزميات الاختزال.

من أمثلة خوارزمية الاختزال هي خوارزمية البحث الثنائية.

3.10.خوارزميات البحث والتعداد

يمكن تصميم العديد من المشكلات (مثل لعب الشطرنج) على أنها مشكلات في الرسوم البيانية.

وتحدد خوارزمية استكشاف الرسم البياني قواعد للتنقل حول الرسم البياني وهي مفيدة لمثل هذه المشاكل.

تشمل هذه الفئة أيضًا خوارزميات البحث والتعداد الفردي وخوارزميات الترابط والتراجع.

4.10. الخوارزمية العشوائية

هذه الخوارزميات تجعل بعض الخيارات عشوائية (أو عشوائية زائفة.(

يمكن أن تكون مفيدة جدًا في إيجاد حلول تقريبية للمشكلات حيث يمكن أن يكون إيجاد الحلول الدقيقة غير عملي. لبعض هذه المشاكل، من المعروف أن أسرع عمليات التقريب يجب أن تتضمن بعض العشوائية. والتساؤل حول ما إذا كانت الخوارزميات العشوائية ذات التعقيد الزمني متعدد الحدود يمكن أن تكون أسرع من خوارزميات أخرى لبعض المشاكل هو سؤال مفتوح يُعرف باسم P مقابل.NP problem

هناك فئتان أساسيتان من هذه النوع وهي:

1.4.10. خوارزميات مونتي كارلو

خوارزمية مونتي كارلو (بالإنجليزية (Monte Carlo :تعطي إجابة صحيحة ذات احتمالية عالية.

2.4.10. خوارزميات لاس فيغاس

خوارزميات لاس فيغاس تُعطي دائمًا الإجابة الصحيحة، ولكن وقت تشغيلها مرتبط بشكل احتمالي فقط.

3.10. خوارزميات الحد من التعقيد

تتضمن هذه التقنية حل مشكلة صعبة من خلال تحويلها إلى مشكلة معروفة بشكل أفضل لدينا والتي نستطيع حلها أو نأمل في حلها باستخدام خوارزميات مناسبة. الهدف هو العثور على خوارزمية اختزال خالية من التعقيدات. على سبيل المثال، تتضمن خوارزمية الاستعلام المستخدمة في العصور على قيمة الوسيط في قائمة غير مرتبة من الأعداد تقوم أولا بفرز قائمة الأعداد

(الجزء الصعب في المهمة) ثم تعيين العنصر الأوسط في القائمة التي تم فرزها (الجزء السهل في المهمة)، تُعرف هذه التقنية أيضًا باسم حوّل تسد (بالإنجليزية.(Transfer and Conquer :

4.10. خوارزميات التتبع العكسي

في هذا النهج، يتم إنشاء حلول متعددة بشكل متزايد ويتم التخلي عنها عندما يتم تحديد أنها لا يمكن أن تؤدي إلى حل كامل صالح.

11) تصنيف الخوارزميات حسب أسلوب التحسين

يوجد تصنيف أكثر تحديدًا للخوارزميات بحسب أسلوب التحسين المستخدم في مشكلات التحسين (بالإنجليزية : Optimization).

وقد تندرج خوارزمية تقوم بحل أحد هذه المشاكل في واحدة أو أكثر من الفئات العامة الموضحة سابقًا وكذلك في إحدى الفئات التالية:

1.11. خوارزميات البرمجة الخطية

عند البحث عن حلول مثالية لوظيفة خطية مرتبطة بقيود المساواة وعدم المساواة الخطية، يمكن استخدام قيود المشكلة مباشرةً في إنتاج الحلول المثلى. هناك خوارزميات يمكنها حل أي مشكلة في هذه الفئة، مثل خوارزمية التبسيط (بالإنجليزية : Simplex)الشائعة. تتضمن المشكلات التي يمكن حلها باستخدام البرمجة الخطية مشكلة التدفق القصوى للرسوم البيانية الموجهة. إذا كانت هناك مشكلة تتطلب أيضًا أن يكون واحدًا أو أكثر من العناصر المجهولة عددًا صحيحًا، يتم تصنيفها في برمجة عدد صحيح. يمكن لخوارزمية البرمجة الخطية حل مثل هذه المشكلة إذا كان من الممكن إثبات أن جميع القيود على القيم الصحيحة سطحية، أي أن الحلول تفي بهذه القيود على أي حال. في الحالة العامة، يتم استخدام خوارزمية متخصصة أو خوارزمية تبحث عن حلول تقريبية، اعتمادًا على صعوبة المشكلة.

2.11. خوارزميات البرمجة الديناميكية

عندما تُظهر المشكلة الهياكل الفرعية المثلى – بمعنى أنه يمكن بناء الحل الأمثل للمشكلة من الحلول المثلى للمشكلات الفرعية – والمشكلات الفرعية المتداخلة، مما يعني أن المشاكل الفرعية نفسها تُستخدم لحل العديد من حالات المشكلة المختلفة، فإن نهجًا أسرع يسمى البرمجة الديناميكية (بالإنجليزية Dynamic Programming) :يتجنب حلول إعادة الحوسبة التي تم حسابها بالفعل.

على سبيل المثال، في خوارزمية(Floyd – Warshall) ، إيجاد أقصر مسار لهدف من قمة في رسم بياني مرجح باستخدام أقصر مسار للهدف من جميع القمم المجاورة. فالبرمجة الديناميكية تتم جنبًا إلى جنب مع التذكر. والفرق الرئيسي بين خوارزميات البرمجة الديناميكية وخوارزميات فرّق تسد هو أن المشاكل الفرعية مستقلة إلى حد ما في خوارزميات فرق تسد، في حين تتداخل المشاكل الفرعية في البرمجة الديناميكية.

يكمن الاختلاف بين البرمجة الديناميكية وخوارزميات التكرار المباشرة في التخزين المؤقت أو التذكر للطلبات التكرارية.

عندما تكون المشاكل الفرعية مستقلة ولا يوجد تكرار، لا يساعد التذكر؛ وبالتالي فإن البرمجة الديناميكية ليست حلا لجميع المشاكل المعقدة. باستخدام التذكير أو الحفاظ على جدول من المشاكل الفرعية التي تم حلها بالفعل، تقلل البرمجة الديناميكية من الطبيعة الأسية لتعقيدات المشكلات إلى تعقيدات ذات طبيعة أقرب لكثيرات الحدود.

3.11. الخوارزميات الجشعة

تشبه الخوارزمية الجشعة (بالإنجليزية Greedy Algorithm) :خوارزمية البرمجة الديناميكية من حيث أنها تعمل عن طريق فحص الهياكل الفرعية، ولكن ليس من المشكلة ولكن من حل معين. تبدأ هذه الخوارزميات بوضع بعض الحلول، والتي يمكن إعطاؤها أو تم إنشاؤها بطريقة ما، وتقوم بتحسينها عن طريق إجراء تعديلات صغيرة. بالنسبة لبعض المشاكل، يمكن العثور على الحل الأمثل المُطلق للمشكلة، بينما بالنسبة لمشاكل أخرى تتوقف هذه الخوارزميات عند حد إيجاد أفضل حل محليًا، أي الحل الذي لا تستطيع الخوارزمية تحسينه ولكنه ليس الحل المثالي المُطلق.

الاستخدام الأكثر شيوعًا للخوارزميات الجشعة هو العثور على الحد الأدنى من شجرة الامتداد حيث يمكن العثور على الحل الأمثل باستخدام هذه الطريقة.

4.11. الطريقة الإرشادية

في مشاكل التحسين، يمكن استخدام الخوارزميات الإرشادية لإيجاد حل قريب من الحل الأمثل في الحالات التي يكون فيها إيجاد الحل الأمثل غير عملي. تعمل هذه الخوارزمية عن طريق الاقتراب من الحل الأمثل كلما تقدمت. من حيث المبدأ، إذا تم تشغيلها لفترة زمنية غير محدودة، فسيجدون الحل الأمثل. تتمتع هذه الخوارزمية بسمة الجدارة، وهي تعني أنه يمكنهم العثور على حل قريب جدًا من الحل الأمثل في وقت قصير نسبيًا. تتضمن هذه الخوارزمية البحث المحلي (بالإنجليزية(Local Search :، والبحث المحظور (بالإنجليزية(Tabu Search :، التلدين المحاكي (بالإنجليزية(Simulated Annealing :، والخوارزمية الجينية (بالإنجليزية.(Genetec Algorithm :

البعض من هذه الخوارزميات، مثل التلدين المحاكي، هي خوارزميات غير حتمية بينما البعض الآخر، مثل البحث المحظور، حتمية.

عندما يتم التعرف على خطأ الحل غير الأمثل، يتم تصنيف الخوارزمية أيضًا على أنها خوارزمية تقريبية.

12) تصنيف الخوارزميات حسب مجال الدراسة

لكل مجال من مجالات العلوم مشاكله الخاصة ويحتاج إلى خوارزميات فعّالة.

غالبًا ما يتم دراسة المشكلات ذات الصلة في مجال واحد معًا.

بعض الفئات هي خوارزميات البحث، الفرز، الدمج، الخوارزميات العددية، خوارزميات الرسم البياني، خوارزميات السلسلة، الخوارزميات الهندسية الحسابية، الخوارزميات التوافقية، الخوارزميات الطبية، خوارزميات التعلّم الآلي، والتشفير، وضغط البيانات، خوارزميات التحليل الإحصائي وخوارزميات تنقيب البيانات.

تميل الحقول إلى التداخل مع بعضها البعض، وقد يحسّن تقدم الخوارزميات في أحد الحقول خوارزميات من الحقول الأخرى، وأحيانًا غير ذات صلة تمامًا بالخوارزمية الأولى. على سبيل المثال، تم اختراع البرمجة الديناميكية لتحسين استهلاك الموارد في الصناعة ولكنها تُستخدم الآن في حل مجموعة واسعة من المشاكل في العديد من المجالات الأخرى.

13) تصنيف الخوارزميات حسب درجة التعقيد

يمكن تصنيف الخوارزميات حسب مقدار الوقت الذي تحتاجه لإكمال المهمة مقارنةً بحجم المدخلات:

الوقت الثابت: إذا كان الوقت الذي تحتاجه الخوارزمية هو نفسه، بغض النظر عن حجم الإدخال. على سبيل المثال الوصول إلى عنصر صفيف.

الوقت اللوغاريتمي: إذا كان الوقت دالة لوغاريتمية لحجم الإدخال. على سبيل المثال خوارزمية البحث الثنائي.

الوقت الخطي: إذا كان الوقت متناسبًا طرديًا مع حجم الإدخال. على سبيل المثال اجتياز قائمة.

وقت كثير الحدود: إذا كان الوقت قوة لحجم المدخلات. على سبيل المثال خوارزمية فرز الفقاعة لها تعقيد زمني تربيعي.

الوقت الأسي: إذا كان الوقت دالة أسية لحجم الإدخال.

على سبيل المثال خوارزمية البحث الشامل أو المعروفة باسم التوليد والاختبار.

قد تحتوي بعض المشكلات على خوارزميات متعددة ذات تعقيد مختلف، بينما قد لا تحتوي المشكلات الأخرى على خوارزميات أو لا توجد خوارزميات فعالة معروفة لحلها. هناك أيضًا تعيينات من بعض المشاكل إلى مشاكل أخرى. وبسبب ذلك، وُجد أنه من الملائم أكثر أن يتم تصنيف المشاكل نفسها بحسب تعقيد أفضل الخوارزميات التي يمكن أن توجد حلا لها بدلا من تصنيف الخوارزميات بحسب تعقيدها.

ملاحظات عامة :

- يمثل التحقق المنهجي خطوة بخطوة لكل من خوارزمياتك أكثر من نصف العمل الذي يتعين إنجازه