مجلة جامعة تشرين للبحوث والدراسات العلمية – سلسلة العلوم الأساسية المجلد (30) العدد (3) 2008
Tishreen University Journal for Research and Scientific Studies - Basic Sciences Series Vol. (30) No. (3) 2008

إنشاء مخطط هاس لشبكات القواسم برمجيا

الدكتور أحمد الغصين*
عبد القادر شيخ إبراهيم**

(تاريخ الإيداع 14 / 4 / 2008. قُبِل للنشر في 2008/6/30)

□ الملخّص □

إن شبكات قواسم عدد صحيح هي من أبرز الأمثلة على الشبكات، لأنها تتنوع تنوعا كبيراً فبعضها يحقق شرط الشبكة التامة، و منها ما يحقق شرط الشبكة التوزيعية، و منها ما يحقق الشرطين و بالتالي يشكل جبر بول.

إن الهدف من هذا البحث هو تقديم خوارزمية برمجية لإنشاء مخطط هاس تفاعلي لبعض شبكات قواسم عدد صحيح، هذه الخوارزمية تولد تلقائيا مخطط شبكة قواسم العدد المطلوب بمجرد إعطائها تحليل هذا العدد إلى عوامله الأولية. حيث إننا لن نخوض في خوارزميات تحليل الأعداد الصحيحة إلى عواملها الأولية وسنأخذ هذه العوامل مباشرة كمعطيات دخل. تُعدّ هذه الخوارزمية خطوة هامة نحو إيجاد تطبيقات برمجية على البنى الجبرية بشكل عام، وعلى دراسة الشبكات بشكل خاص. لأن مخطط هاس له فائدة كبيرة في تبيان طبيعة وخواص الشبكة التي يمثلها.

الكلمات المفتاحية: شبكات قواسم عدد صحيح، مخطط هاس.

* أستاذ – قسم الرياضيات– كلية العلوم–جامعة تشرين – اللاذقية – سورية.
** طالب ماجستير – قسم الرياضيات شعبة الجبر – كلية العلوم–جامعة تشرين – اللاذقية – سورية.

مجلة جامعة تشرين للبحوث والدراسات العلمية – سلسلة العلوم الأساسية المجلد (30) العدد (3) 2008
Tishreen University Journal for Research and Scientific Studies - Basic Sciences Series Vol. (30) No. (3) 2008

Programmatically-Creating Hass Diagram of Divisors Lattices

Dr. Ahmad Alghoussein *
Abdul Kader Ibrahim **

(Received 14 / 4 / 2008. Accepted 30/6/2008)

□ ABSTRACT □

The lattices of divisors of an integer are one of the most important examples on lattices, there are many kinds of it, some are completed, some are distributed, and some other form Boolean algebra. In this paper we will give a program algorithm to create an interactive Hass Diagram of some of divisors lattices. This algorithm will generate automatically the diagram of the lattice of divisors of a given integer. The integer number must be given as product of its prime factors, because we will not talk about factoring algorithms, and we will input those factors manually.

This algorithm is an important step to develop software applications on algebraic structures, especially on lattices, because of the great importance of Hass diagram in the study of lattices theory.

Keywords: Lattices of divisors, Hass Diagram.

* **Prof., Dep. of Math. Faculty of science, Tishreen University Lattakia Syria.**
** **Postgraduate Student, Dep. of Math. Faculty of science, Tishreen University Lattakia Syria.**

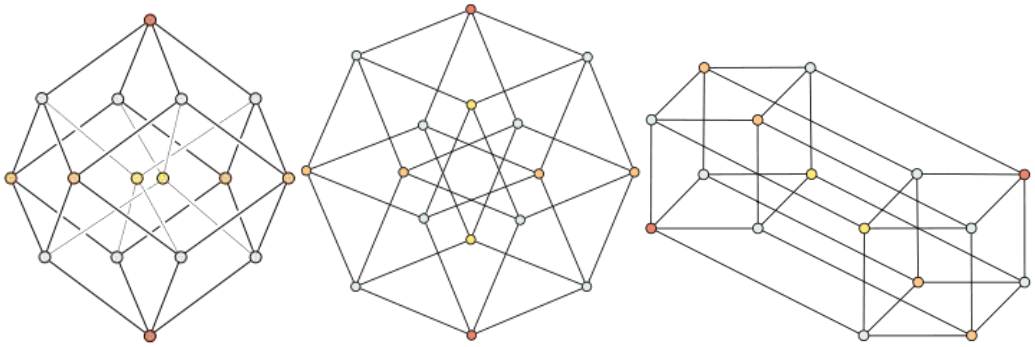
مقدمة:

مع تطور الجبر العام في فترة 1930 على يد غرات بيخوف و أوستين أور وآخرين، أصبحت نظرية الشبكات موضوعاً فعالاً ونامياً في كل من تطبيقاتها الجبرية و مسائلها المشوقة المستقلة. و أصبح رسم مخطط هاس للشبكة خطوة هامة لدراستها، فإذا نظرنا إليه و لم نجد فيه أي شبكة جزئية على شكل شبكة البنتاغون الخماسية فالشبكة معيارية، و إذا لم نجد أي واحدة على شكل الشبكة الماسية فالشبكة توزيعية، وبنظرة سريعة عليه يمكننا معرفة الحدين الأعلى الأصغري و الأدنى الأعظمي لأي عنصرين من الشبكة.

هدف البحث و أهميته:

إن أغلب برامج الحاسوب الخاصة بحل مسائل الرياضيات متخصصة بفرع التحليل، و البرامج الخاصة بحل المسائل الجبرية لا تعدو عن كونها في معظم الأحيان تطبيقات على الجبر الخطي. لذلك نريد كتابة برنامج خاص بحل بعض مسائل الجبر المجرد، و يستغل قدرة الحاسوب الرسومية، لذلك اخترنا دراسة مخطط هاس لشبكات القواسم بأسلوب برمجي.

يمكن رسم مخطط هاس لشبكة منتهية بعدة أشكال، و كل شكل يبرز أسلوب لكتابة الشبكة كجداء لشبكات جزئية فيها، ويبين (الشكل 1) ثلاثة أشكال تمثل مخطط هاس لإحدى الشبكات.



الشكل 1: ثلاثة أساليب لرسم مخطط هاس لنفس الشبكة

إن رسم مخطط هاس للشبكات الصغيرة هو أمر يسير، لكن الأمر يزداد تعقيدا عند التعامل مع شبكات ذات عدد كبير من العناصر. من هنا تبرز الحاجة لوجود برنامج يولد تلقائيا المخطط المطلوب، و يُمكّن المستخدم من تحريك عناصر الشبكة مع المحافظة على العلاقات بينها، و تحديد شبكة جزيئة أو مرشحة أو مثالي ضمن الشبكة الأصلية بلون مختلف، أي يولد مخطط تفاعلي يساعد على دراسة الشبكة و يوفر الوقت اللازم لذلك.

إن ما ينطبق على شبكة قواسم عدد صحيح ينطبق على باقي الشبكات، و وقع اختيارنا عليها لقدرة الحاسوب على التعامل مع الأعداد الصحيحة و علاقة قابلية القسمة بينها بصورة سريعة و أسلوب مباشر.

طرائق البحث و مواده:

تعريف: من المعلوم أن الشبكة المرتبة[1] (Ordered Lattice) هي مجموعة مرتبة جزئياً يوجد فيها لكل زوج من عناصرها حد أعلى أصغري و حد أدنى أعظمي.

ترميز: إذا كانت L شبكة مرتبة فإننا نرمز للحد الأدنى الأعظمي للعنصرين x و y بالرمز x.y و للحد الأعلى الأصغري لهما بالرمز x + y.

تعريف: نقول عن المجموعة الجزئية غير الخالية I من الشبكة L أنها مثالي[2] (Ideal) إذا كان:

1- $x \leq y , y \in I \Rightarrow x \in I$

2 - $x , y \in I \Rightarrow x + y \in I$

تعريف: إن المرشحة[2] (Filter)هي مجموعة جزئية غير خالية F من شبكة L تحقق الشرطين:

1- $x \leq y , x \in F \Rightarrow y \in F$

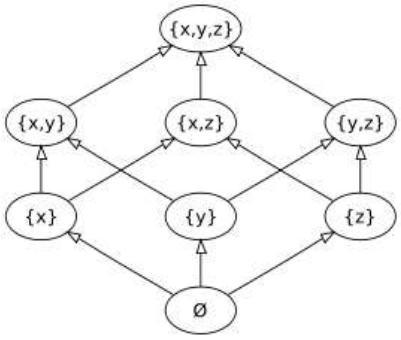
2 - $x , y \in F \Rightarrow xy \in F$

تمثيل الشبكة وفق هاس: لتكن $(E,\leq)$ مجموعة مرتبة جزئيا(Partial ordered set)، لإنشاء مخطط هاس لعلاقة الترتيب $\leq$ على E، فإننا نمثل عناصر E بنقاط أو دوائر صغيرة وإذا كان $x,y \in E$ يحققان الشرطين $x \leq y$ و $x \neq y$ حيث إذا وجد $z \in E$ يحقق المتراجحة $x \leq z \leq y$ فإن $x = z$ أو $x = y$ عندئذ نضع x في الأسفل و y في الأعلى و نصل بينهما بخط مستقيم.

أمثلة على الشبكات:

– لتكن E مجموعة ما، P(E) أسرة المجموعات الجزئية في المجموعة E، عندئذ تشكل $(P(E),\subseteq)$ شبكة مرتبة بالنسبة لعلاقة الاحتواء.

و إذا أخذنا المجموعة {x,y,z} فإن شبكة المجموعات الجزئية فيها تعطى كما في (الشكل 2).



الشكل 2: مخطط هاس لشبكة المجموعات الجزئية في المجموعة {x,y,z}

– لتكن N مجموعة الأعداد الطبيعية عندئذ تشكل $(N,|)$ شبكة مرتبة بالنسبة لعلاقة قابلية القسمة، وهي شبكة لا تحوي عنصراً أكبر، أي هي شبكة غير محدودة.

شبكة القواسم(Lattice of divisors): ليكن n عدد صحيح معطى، و لتكن D(n) مجموعة كل الأعداد الموجبة i حيث $i \leq n$ و التي تحقق الشرط $i|n$، عندئذ تشكل $(D(n),|)$ شبكة مرتبة ندعوها بشبكة قواسم العدد n.

دراسة مخطط الشبكة:

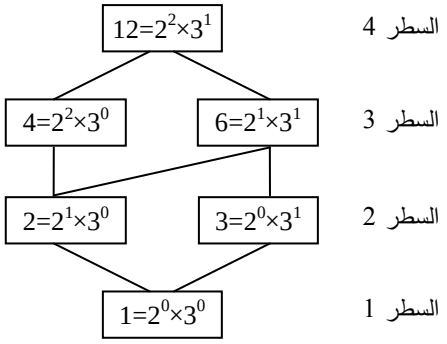
بدراستنا للعديد من شبكات القواسم و برسم مخطط هاس الذي يمثلها يدويا وجدنا ثلاثة أمور تميز مخطط شبكة قواسم أي عدد صحيح:

– بما أن عناصر الشبكة هي قواسم للعدد الصحيح المعطى فإن عواملها الأولية هي نفس عوامل العدد المعطى لكن بقوى أصغر أو تساوي قوى العوامل الأولية للعدد الصحيح و أكبر أو تساوي الصفر، أي يمكن الحصول على جميع قواسم العدد بأخذ جميع الأعداد التي عواملها الأولية تحقق الشرط السابق.

– إن عناصر شبكة القواسم تتوضع على أسطر عددها يساوي مجموع قوى العوامل الأولية للعدد المعطى زائد واحد، بحيث إذا رقمنا الأسطر من الأسفل إلى الأعلى، فإن كل سطر يحوي القواسم التي مجموع قوى عواملها الأولية يساوي رقم السطر ناقص واحد.

– إن عدد الخطوط التي تصل بين عنصر من سطر ما مع عناصر السطر الذي تحته يساوي عدد عوامله الأولية التي قواها أكبر من الصفر.

كمثال بسيط على ذلك لنأخذ الشبكة (|,D(12)) كما في (الشكل 3).



الشكل 3: المخطط التفصيلي لشبكة قواسم العدد 12

نلاحظ أن السطر 1 يحوي الأعداد التي مجموع قوى عواملها الأولية صفر، أي يحوي العدد 1 فقط، و السطر 2 الذي يعلوه يحوي الأعداد التي مجموع قوى عواملها الأولية يساوي 1، أي يحوي العددين 2 ، 3. أما السطر 3 فيحوي الأعداد التي مجموع قوى عواملها الأولية يساوي 2 أي يحوي العددين 4 ، 6. وهكذا حتى الوصول إلى السطر الأول من الأعلى فيه جميع العناصر التي مجموع قوى عواملها الأولية يساوي مجموع قوى العوامل الأولية للعدد الذي نرسم شبكة قواسمه، أي يحوي فقط العدد 12.

ومن جهة أخرى، إن عدد العوامل الأولية للعددين 6 ، 12 التي قواها أكبر من الصفر هو 2 لهذا يرتبط كل منهما مع السطر الذي تحته خطين، بينما يرتبط العدد 4 مع السطر الذي تحته خط واحد لأن له عامل أولي واحد قوته أكبر من الصفر، و الأمر ذاته ينطبق على الأعداد 2 ، 3.

النتائج و المناقشة:

تصميم واجهة البرنامج:

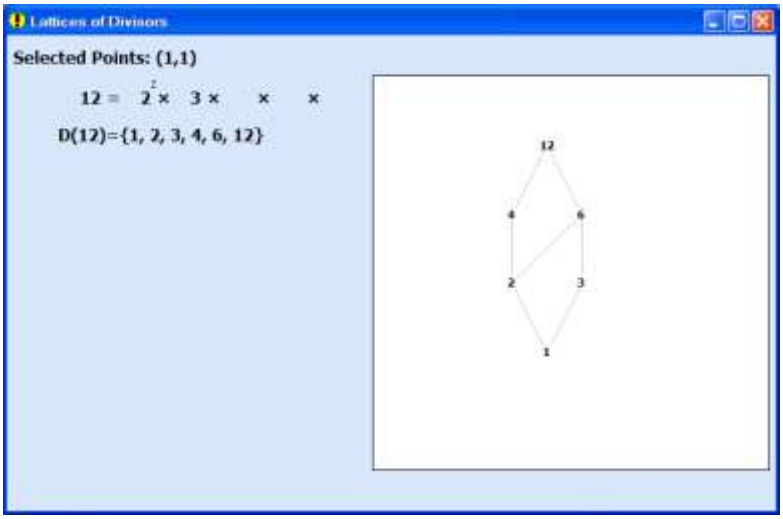
سنستعمل لوضع البرنامج المطلوب لغة فيجوال بيسك الإصدار السادس لأنها تؤمن جميع الأدوات التي نحتاجها لبناء البرنامج المطلوب.

في أول مرحلة من مراحل تصميم البرنامج، وضعنا على النافذة الرئيسية عشرة مربعات نص (TextBox[3]) خمسة منها لإدخال عوامل العدد الأولية، و وضعنا فوق كل مربع نص منها مربع آخر لإدخال قوة كل عامل، صغرنا حجم الخط فيه، وغيرنا لون خلفية (BackColor[3]) كل الصناديق ليماثل لون النافذة حيث تبدو هذه الصناديق مدموجة بالنافذة. ثم فصلنا بين المربعات الخمسة الأولى بأربع لافتات (Label[3]) ضبطنا خاصية العنوان لها لتظهر الرمز ×، و وضعنا قبل مربع النص الأول أداة لافتة لنضع فيها العدد الناتج عند تغيير نص أحد مربعات النصوص

العشرة السابقة، و وضعنا مربع لوحة (PictureBox[3]) يعطي مساحة بيضاء تشغل معظم الشاشة لإنشاء المخطط فيها.

بما أن المخطط مؤلف من مجموعة من الأعداد تصل بينها مجموعة من الخطوط، استخدمنا لإنشاء مخطط الشبكة أداة لافتة (Label[3]) واحدة وضعناها على مربع اللوحة السابق، وضبطنا خاصية الفهرس (Index[3]) بالعدد 0، ووضعنا مع أداة اللافتة أداة خط (Line[3]) واحدة ضبطنا أيضا في خاصية الفهرس بالعدد 0. أي أننا شكلنا مصفوفتي (Array) أدوات. سنستعمل اللافتة لتمثيل عناصر الشبكة و الخط لتمثيل العلاقة بين العناصر.

لا بد من الإشارة إلى أنه يمكن استخدام إجراءات الرسم المبيتة ضمن فيجوال بيسيك، لكن هدفنا إنشاء مخطط تفاعلي يُمكّن المستخدم من تحريك العناصر عن طريق سحبها و إفلاتها، و تحديد شبكة جزئية أو مثالي أو مرشحة بتغيير لون الخطوط الواصلة بين العناصر، وبما أن اللافتة تستجيب لأحداث الفأرة من نقر وسحب وإفلات، و بما أن مصفوفات الأدوات تساعد على كتابة إجراء يتعامل مع جميع عناصرها، و يمكن زيادة عدد هذه العناصر أو إنقاصه حسب الحاجة، استعملنا الأدوات بدلاً من إجراءات الرسم. كما لابد أيضا من الإشارة إلى أن مربعات النصوص السابقة تسمح بإدخال خمس عوامل أولية فقط للعدد المعطى، ويمكن زيادة عدد هذه المربعات عند الحاجة.



الشكل 4: واجهة البرنامج بعد تشغيله و إعطائه الأمر برسم شبكة قواسم العدد 12

كتابة الشيفرة البرمجية:

عند تغيير النص الموجود في أحد مربعات النصوص يقع الحدث Change [3]، ضمن إجراء هذا الحدث نضع عبارة لحساب العدد الناتج عن ضرب العوامل الأولية المدخلة في صناديق النصوص الخمسة السفلية بعد رفعها إلى القوى المدخلة في الصناديق العلوية، و لتخزين العدد مع عوامله الأولية نعرف نمطاً (Type[3]) جديداً بالشكل التالي:

```
Private Type Primes
    Prime(4) As Integer
    Expo(4) As Integer
    Number As Long
    Es As Integer
End Type
```

نخزن في المصفوفة Prime(4) العوامل الأولية الخمسة (ترقيم عناصر المصفوفة يبدأ من 0 لذلك وضعنا4)، وفي المصفوفة Expo(4) قوى العوامل الأولية بنفس الترتيب، وفي Number العدد نفسه، أما في Es فنخزن مجموع القوى السابقة لحاجتنا إليه في الخوارزمية كما و للاستغناء عن حسابه كلما احتجنا إليه لاحقاً.

و بعد ذلك نعلن عن المتغيرات العامة المطلوبة كما يلي:

```
Dim Num As Primes
Dim Divisors() As Primes
Dim Es() As Integer
Dim Lines As Integer
```

المتغير Num يمثل العدد وعوامله الأولية و يتم حسابه ضمن الحدث Change لمربعات النصوص العشرة، المصفوفة Divisors() هي مصفوفة ديناميكية (يمكن تحديد عدد عناصرها حسب الحاجة) يمثل كل عنصر فيها أحد قواسم العدد المعطى مع عوامله الأولية، أما المصفوفة Es() فهي مصفوفة ديناميكية نحدد عدد عناصرها ليساوي عدد أسطر المخطط المطلوب، و يمثل العنصر ذو الترتيب i فيها عدد القواسم التي مجموع قوى عواملها الأولية هو العدد i، وبالنسبة للمتغير Lines فهو يمثل عدد العلاقات (الخطوط) اللازم تمثيلها.

و على ضوء ما أوردناه سابقا في الفقرة 3، فإننا نكتب الإجراء الأساسي الذي يحسب قواسم العدد والمعلومات اللازم تخزينها في المتغيرات سابقة الذكر:

```
Private Sub GenerateDivisors()
Dim h%, i%, j%, k%, m%, l%

ReDim Divisors(0)  'نضبط حجم المصفوفة لتحوي عنصرا واحدا فقط (ترقيم المصفوفة يبدأ من 0)
ReDim Es(Num.Es)   'نضبط حجم المصفوفة ليساوي مجموع قواسم العدد المعطى زائد واحد أي يساوي عدد أسطر المخطط
Lines = 0          'نهيئ المتغير بالعدد 0

For l = 0 To Num.Es       'نكرر من أول سطر في المخطط إلى آخر سطر (نبدأ من 0 وحتى عدد الأسطر ناقص 1)
For m = 0 To Num.Expo(4)  'نولد القواسم عن طريق أخذ العوامل الأولية للعدد المعطى
For k = 0 To Num.Expo(3)  'وإعطائها قوى تبدأ من 0 و تنتهي بقوة العامل في العدد المعطى
For j = 0 To Num.Expo(2)  'ولأنه لدينا خمس عوامل أولية نكتب خمس تكرارات متداخلة
For i = 0 To Num.Expo(1)
For h = 0 To Num.Expo(0)
  If h + i + j + k + m = l Then 'إذا كان مجموع القوى يساوي رقم السطر ناقص 1
    'عوامل القاسم هي نفس عوامل العدد المعطى
    Divisors(UBound(Divisors)).Prime(0) = Num.Prime(0)
    Divisors(UBound(Divisors)).Prime(1) = Num.Prime(1)
    Divisors(UBound(Divisors)).Prime(2) = Num.Prime(2)
    Divisors(UBound(Divisors)).Prime(3) = Num.Prime(3)
    Divisors(UBound(Divisors)).Prime(4) = Num.Prime(4)
    'قوى العوامل الأولية نأخذها من التكرار
    Divisors(UBound(Divisors)).Expo(0) = h
    Divisors(UBound(Divisors)).Expo(1) = i
    Divisors(UBound(Divisors)).Expo(2) = j
    Divisors(UBound(Divisors)).Expo(3) = k
    Divisors(UBound(Divisors)).Expo(4) = m
    'نحسب القاسم بأخذ جداء العوامل الأولية بعد رفعها إلى قواها
```

```
    Divisors(UBound(Divisors)).Number = _
      (Num.Prime(0) ^ h) * (Num.Prime(1) ^ i) * _
      (Num.Prime(2) ^ j) * (Num.Prime(3) ^ k) * _
      (Num.Prime(4) ^ m)
    نحسب مجموع القوى'
    Divisors(UBound(Divisors)).Es = h + i + j + k + m
    نضيف 1 إلى عدد العوامل الأولية التي مجموع قواها يساوي رقم السطر ناقص 1'
    Es(l) = Es(l) + 1
    نضيف إلى عدد الخطوط الكلي عدد الخطوط التي يرتبط بها هذا العنصر مع عناصر السطر الذي تحته'
    Lines = Lines + Sgn(h) + Sgn(i) + Sgn(j) + _
                    Sgn(k) + Sgn(m)
    نضيف عنصر جديد إلى المصفوفة'
    ReDim Preserve Divisors(UBound(Divisors) + 1)
  End If
Next
Next
Next
Next
Next
Next
بعد الانتهاء يكون في نهاية المصفوفة عنصر فارغ زائد نحذفه'
ReDim Preserve Divisors(UBound(Divisors) - 1)
End Sub
```

إن الإجراء السابق يولد قواسم العدد عن طريق أخذ العوامل الأولية للعدد و إعطائها قوى متزايدة من الصفر و حتى أكبر قوة للعامل الأولي تقسم العدد. أما لحساب المتغير Lines قلنا أن كل قاسم في الشبكة يكون عدد العناصر التي ترتبط به من الأسفل يساوي عدد العوامل الأولية لهذا العنصر التي قواها تختلف عن الصفر. فإذا نظرنا إلى السطر

```
Lines = Lines + Sgn(h) + Sgn(i) + Sgn(j) + Sgn(k) + Sgn(m)
```

لوجدنا أنه يستخدم التابع Sgn [3] لتحديد إذا كانت القوة تساوي الصفر أم لا، معلوم أن التابع السابق يعيد 1- إذا كان العدد سالبا و 1 إذا كان العدد موجبا و 0 إذا كان العدد صفرا، فبدلا من استخدام خمس عبارات شرطية قمنا باستخدام هذا التابع.

الآن نكتب الأجراء الذي يقوم بإنشاء المخطط، و مهمته تتلخص بضبط عدد مصفوفة اللافتات و مصفوفة الخطوط ثم تحريك كل عنصر من عناصر المصفوفتين إلى موقعه المناسب، و من أجل ضبط المسافة بين عناصر الشبكة أعلنا عن الثابتين التاليين:

```
Const SPA = 1000    المسافة بين كل عنصرين طوليا أو عرضيا'
Const SPR = 2       عامل تقسيم يحدد مسافة القفزة'
```

فبتعديل هذين الثابتين يمكن تعديل المسافة بين عناصر المخطط بدون تعديل الشيفرة البرمجية للأجراء الخاص بتوليد مخطط هاس لشبكة القواسم المطلوبة و الذي نكتبه على النحو التالي:

```
Private Sub GenerateLattice()
Dim i%, j%, l%, Esp() As Integer
Dim LastS As Integer
نضيف اللافتات اللازمة و نزيل الزائدة عن حاجتنا'
```

```
For i = lblP.Count To UBound(Divisors)
    Load lblP(i)
Next
For i = UBound(Divisors) + 1 To lblP.Count - 1
    Unload lblP(i)
Next
نضيف الخطوط اللازمة ونزيل الزائدة عن حاجتنا '
For i = linL.Count To Lines - 1
    Load linL(i)
Next
For i = Lines To linL.Count - 1
    Unload linL(i)
Next
عدد عناصر المصفوفة يساوي عدد الأسطر، كل عنصر من هذه المصفوفة يمثل عدد عناصر السطر التي تم ضبط موضعها '
ReDim Esp(Num.Es)
تهيئة المتغيرات بالعدد 0 '
LastS = 0   هذا المتغير يمثل آخر عنصر من عناصر السطر الذي تحت العنصر الحالي '
l = 0
For i = 0 To lblP.Count - 1   نكرر من أول لافتة إلى آخر لافتة '
    Dim nX%, nY%
نحسب عدد القفزات العرضية إبتداءً من حافة مربع اللوحة اليسرى و حتى العنصر المطلوب اعتمادا على عدد عناصر السطر الكلية و عدد العناصر التي تم ضبط موقعها '
    nX = Int((picRight.ScaleWidth / 2 + SPA / 2 - _
         Es(Divisors(i).Es) / 2 * SPA + _
         Esp(Divisors(i).Es) * SPA) / (SPA / SPR))
نحسب عدد القفزات الطولية إبتداءً من الحافة العليا لمربع اللوحة و حتى السطر الذي ينتمي إليه العنصر إعتمادا على عدد الأسطر الكلية و على رقم السطر الذي ينتمي إليه العنصر '
    nY = Int((picRight.ScaleHeight / 2 + _
         Num.Es / 2 * SPA - _
         Divisors(i).Es * SPA) / (SPA / SPR))
    نضبط عنوان اللافتة لتحوي القاسم '
    lblP(i).Caption = Trim(Divisors(i).Number)
    lblP(i).Tag = Trim(Divisors(i).Number)
    نحرك اللافتة إلى موقعها المناسب اعتمادا على عدد القفزات الطولية و العرضية '
    lblP(i).Move nX * (SPA / SPR) - lblP(i).Width / 2, _
                 nY * (SPA / SPR) - lblP(i).Height / 2
    نضيف 1 إلى عدد عناصر السطر التي تم ضبط موقعها ثم نظهر اللافتة '
    Esp(Divisors(i).Es) = Esp(Divisors(i).Es) + 1
    lblP(i).Visible = True
    إذا كان مجموع قوى العنصر مختلف عن الصفر فالعنصر مرتبط مع عناصر السطر الذي تحته '
    If Divisors(i).Es Then
      نكرر من أول عنصر من عناصر السطر الذي يقع تحت العنصر الحالي إلى آخر عنصر '
      For j = LastS - Es(Divisors(i).Es - 1) + 1 To LastS
        نختبر علاقة يقسم '
        If Divisors(i).Number Mod Divisors(j).Number = 0 Then
          العلاقة محققة ، نحرك أحد الخطوط ليصل بين العنصرين '
```

```
            linL(l).X1 = lblP(j).Left + lblP(j).Width / 2
            linL(l).Y1 = lblP(j).Top + lblP(j).Height / 2
            linL(l).X2 = lblP(i).Left + lblP(i).Width / 2
            linL(l).Y2 = lblP(i).Top + lblP(i).Height / 2
            linL(l).Visible = True
            linL(l).Tag = j & "-" & i  'نخزن رقمي العنصرين الذين يصل بينهما الخط لحاجتنا إليه عند تحريك أحد العنصرين
            l = l + 1
          End If
        Next
        'إذا كان هذا العنصر آخر عنصر في السطر فاحتفظ برقمه لحاجتنا إليه
        If LastS + Es(Divisors(i).Es) = i Then LastS = i
      End If
      lblP(i).ZOrder  'إجعل ترتيب اللافتة فوق ترتيب باقي الأدوات
Next  'انتقل إلى العنصر التالي
End Sub
```

إن lblP() هي مصفوفة اللافتات التي نستخدمها لتمثيل عناصر الشبكة، و linL() هي مصفوفة الخطوط التي نستخدمها لتمثيل العلاقات بين العناصر، و picRight هو مربع اللوحة الذي يحوي مصفوفتي اللافتات و الخطوط، كما أسلفنا في الفقرة 4.

في بداية الإجراء نقوم بالتخلص من اللافتات و الخطوط الزائدة عن طلبنا إذا كان هنالك زيادة، أما إذا لم توجد زيادة نقوم بإنشاء ما يلزمنا من لافتات و خطوط، وذلك خلال حلقات التكرار الأربع الأولى. تجدر الإشارة أنه يلزمنا عدد من اللافتات يساوي عدد العناصر في المصفوفة Divisors()، وعدد من الخطوط يساوي العدد الموجود في المتغير Lines، تم حساب هذه المصفوفة و هذا المتغير في الإجراء GenerateDivisors الذي أسلفنا ذكره.

لترتيب العناصر بالشكل الصحيح يبرز استخدام المصفوفة Es() التي سبق ذكرها، فمن خلالها نحدد عدد العناصر في أي سطر. الآن ندخل في تكرار لضبط موقع و نص كل من اللافتات، فنضع في المتغيرين nX و nY عدد القفزات الطولية و العرضية التي تحدد إحداثيات العنصر الحالي معتمدين على عدد العناصر في السطر الذي ينتمي إليه (موجود في المصفوفة Es) وعلى عدد العناصر التي تم ضبط موقعها قبله في نفس السطر (موجود في المصفوفة Esp)، ثم نحرك اللافتة إلى المكان المطلوب عن طريق ضرب عدد القفزات بطول القفزة لنحصل على الإحداثيات، ونضبط عنوانها ليحوي العنصر الحالي.

قلنا سابقا أن كل عنصر في الشبكة يرتبط من الأسفل بعدد من العناصر يساوي عدد عوامله الأولية التي قواها مختلفة عن الصفر، لذلك نختبر القيمة Divisors(i).Es فإذا حوت قيمة مختلفة عن الصفر فمجموع العوامل مختلف عن الصفر و بالتالي أحد العوامل قوته مختلفة عن الصفر، فندخل في تكرار و نختبر جميع عناصر السطر الذي يقع تحت سطر العنصر الحالي لنرى أي منها يقسم العنصر الحالي، ثم نقوم بتحريك أحد عناصر المصفوفة linL() ليصل بين العنصر الحالي و العنصر الذي يقسمه. نقوم أخيرا بنقل اللافتة إلى أمام باقي الأدوات عن طريق الإجراء ZOrder [3] لنضمن ألّا يغطيها أي خط من خطوط المصفوفة linL().

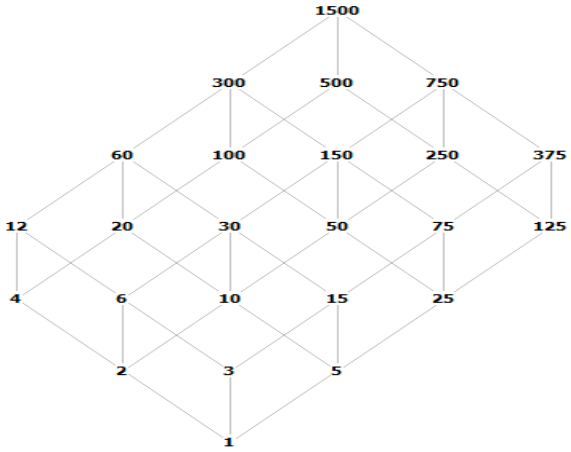
نقوم أخيراً بكتابة إجراءات الأحداث MouseDown، MouseMove، MouseUP الخاصة بمصفوفة اللافتات lblP() لنقل اللافتة إلى موقعها الجديد عند سحبها و إفلاتها، وتحريك كل الخطوط التي تصل إليها إلى الموقع الجديد.

الاستنتاجات و التوصيات:

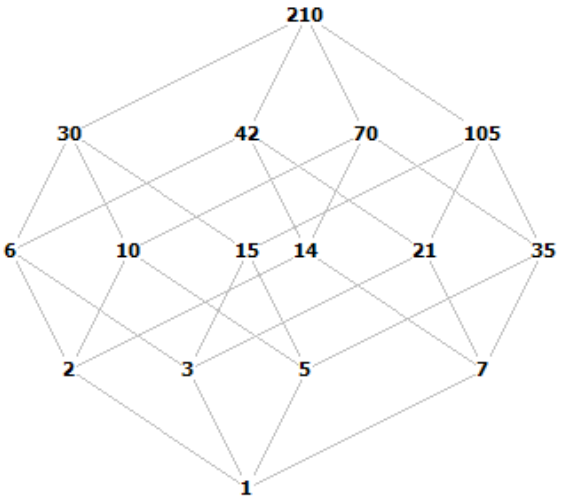
عند اختبار البرنامج السابق حصلنا على نتائج جيدة عند رسم مخططات لشبكات معقدة، و في نهاية البحث نورد بعض المخططات التي حصلنا عليها باستخدام هذا البرنامج. كما أن إمكانية تحريك عناصر المخطط تتيح تحسين شكله لإبراز جانب معين منه، مما يزيد من الفائدة العلمية و التطبيقية للبرنامج.

لكن لا بد من تطوير البرنامج ليقبل إدخال عدد أكبر من العوامل الأولية ، ثم الانتقال من إنشاء مخططات شبكات القواسم إلى إنشاء مخططات أي شبكة منتهية، لتصبح الفائدة عامة على جميع أنواع الشبكات.

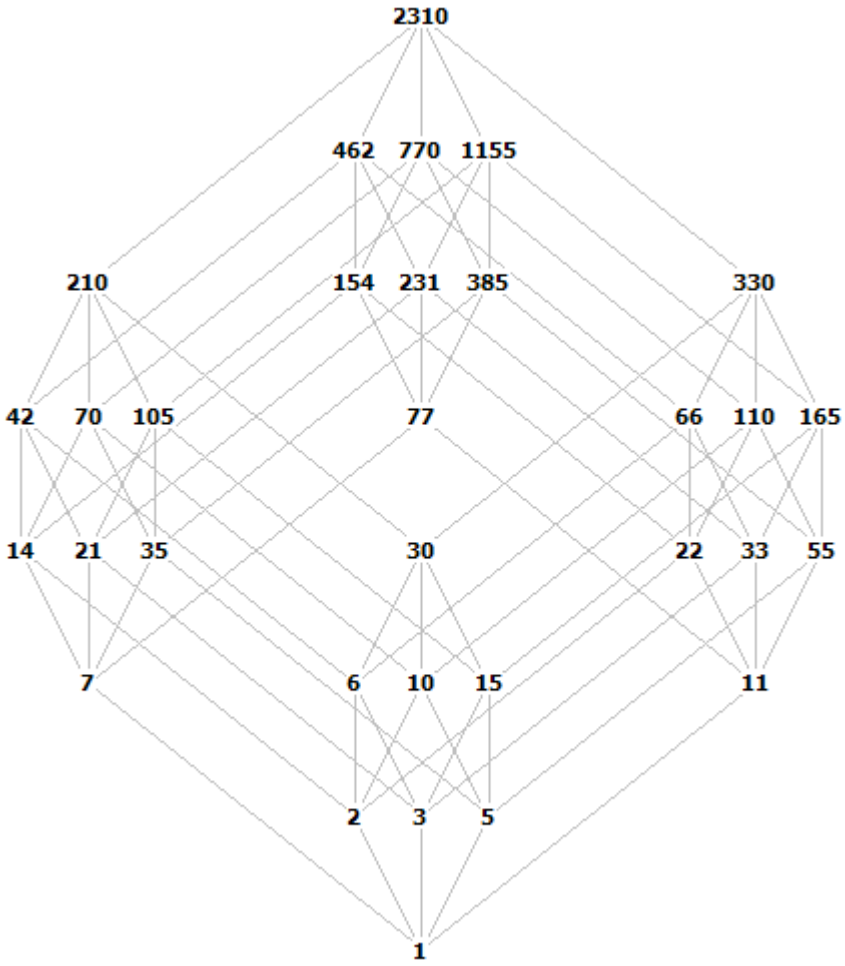
بعض الشبكات التي حصلنا عليها باستخدام هذا البرنامج:



الشكل 5: مخطط هاس لشبكة قواسم العدد 1500



الشكل 6: مخطط هاس لشبكة قواسم العدد 210



الشكل 7: مخطط هاس لشبكة قواسم العدد 2310

المراجع:

[1]- LIDL, R. and PILZ, G. – *Applied Abstract Algebra*, Springer Verlag, New York, 1984.

[2]- BALBES, R. and DWINGER, P. – *Distributive Lattices*, University of Missouri Press, 1974.

[3]– عودة، أيمن و منصور، نوال – *Visual Basic دليل اللغة*، شعاع للنشر والعلوم، حلب، 1998.