The Mathematica® Journal

Improving the Kruskal–Katona Bounds for Complete Subgraphs of a Graph

Robert Cowen

An important problem in graph theory is to find the number of complete subgraphs of a given size in a graph. If the graph is very large, it is usually only possible to obtain upper bounds for these numbers based on the numbers of complete subgraphs of smaller sizes. The Kruskal–Katona bounds are often used for these calculations. We investigate these bounds in specific cases and study how they might be improved.

Introduction

Graph theory has many interesting problems that lend themselves to computer investigation. Mathematica has many graph theory functions that can enable these investigations. We shall introduce the reader to Mathematica's graph theory capability while investigating a problem in extremal graph theory. Extremal graph theory tries to find graphs satisfying certain extreme properties—for example, having the most triangles for a fixed number of edges.

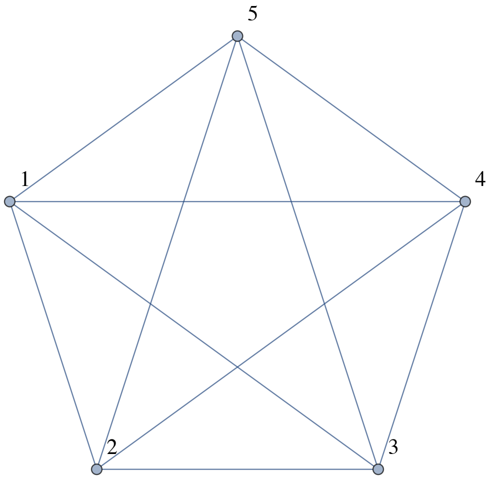
We first introduce some basic graph theory concepts and show how to represent them in Mathematica. Formerly, most graph theory functions were contained in the Combinatorica package; however, this graph theory functionality has, for the most part, been absorbed by the main program, making it unnecessary to load this package.

Graph Theory Basics

A finite graph g consists of two finite sets, V and E. The elements of the set V are called *vertices* and the elements of the set E consist of (unordered) pairs of vertices called *edges*. We often write $g = (V, E)$. A graph $g_1 = (V_1, E_1)$ is called a *subgraph* of graph g if $V_1 \subset V$ and $E_1 \subset E$; that is, if each vertex in the subgraph g_1 is also a vertex in the graph g and each edge of the subgraph g_1 is also an edge of the graph g.

Graphs are often depicted as points (the vertices) and line segments (the edges) that join pairs of vertices in E. Thus, to draw the graph g consisting of the five labeled vertices $V = \{1, 2, 3, 4, 5\}$ and with edge set E being all pairs of vertices, we enter the following command.

```
g = CompleteGraph[5, VertexLabels → Automatic]
```

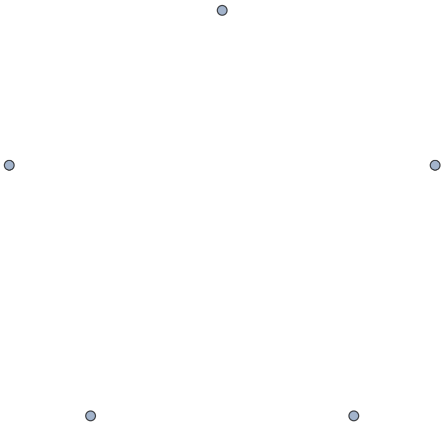


This graph is known as the *complete graph on five vertices* and denoted by K_5.

In general, K_n denotes the complete subgraph on n vertices, that is, the graph with vertex set $\{1, 2, 3, \ldots, n\}$ and edge set consisting of all pairs of elements of V.

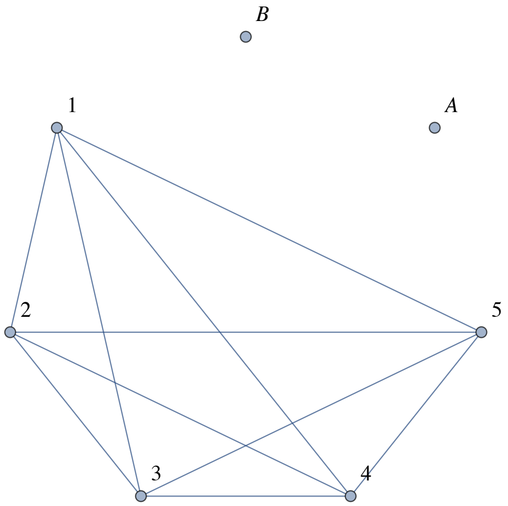
The *complement* of the graph $g = (V, E)$ is the graph having the same set of vertices V and whose edges are exactly those pairs of vertices of V that *do not* belong to E. Thus, the graph complement of the complete graph $g = K_5$ has no edges at all.

```
GraphComplement[g]
```



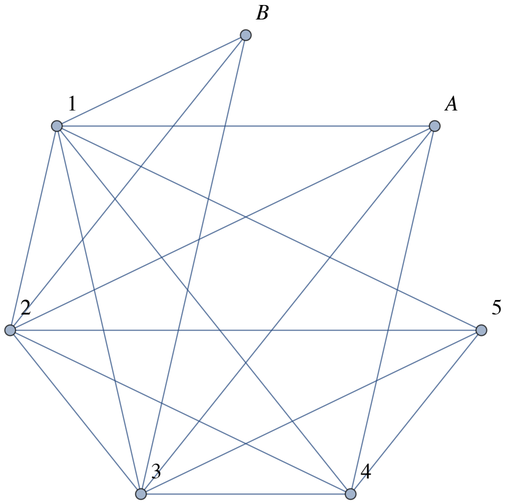
Mathematica can also add vertices to an already existing graph. This adds two vertices labeled A and B to the graph g.

```
g1 = VertexAdd[g, {A, B}]
```



This is how to add edges to graph g_1. (The symbol "⧟" can be entered from the keyboard using the Esc key; press Esc, type u and e, then press Esc again.)

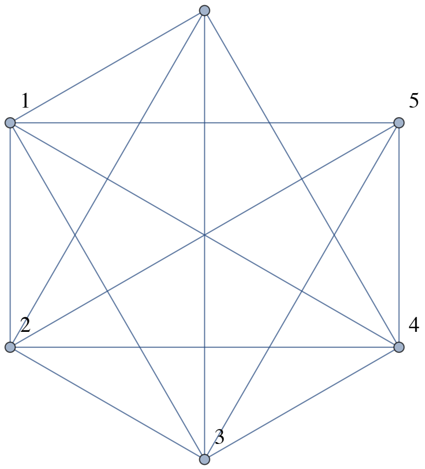
```
g2 = EdgeAdd[g1, {A ⧟ 1, A ⧟ 2, A ⧟ 3, A ⧟ 4, B ⧟ 1, B ⧟ 2,
    B ⧟ 3}]
```



Vertices and edges can be deleted with the commands `VertexDelete` and `EdgeDelete`.

Another useful operation, `VertexContract`, contracts a set of vertices into one vertex. For example, this contracts the graph g_2 by contracting the vertices labeled A and B into a single vertex.

```
VertexContract[g2, {A, B}]
```



□ The Number of Complete Subgraphs of a Graph

An important problem in graph theory is to find the number of complete subgraphs (or cliques) of a graph. For example, find the number of cliques of a certain size in a large social network graph. Here, the people are the vertices, an edge joins two people who know each other and a clique consists of people who all know each other; that is, they form a complete subgraph.

There is a dual version of this problem, equally important, that asks for the number of independent sets of a graph. A set of vertices is *independent* in the graph g if there are *no* edges of g connecting the vertices in the set. (Clearly, a set of vertices of g is independent if and only if they form a complete graph in the complement of g.)

We consider next how to compute the number of complete subgraphs exactly for small graphs and how one might obtain useful upper bounds for larger graphs.

Given K_7, it is easy to determine how many complete subgraphs there are having, say, four vertices; the answer is the number of ways to choose four of the seven vertices, since for each such choice, all edges between the chosen vertices are also present in the original graph. The number of ways to choose four out of seven objects is just the binomial coefficient $\binom{7}{4}$.

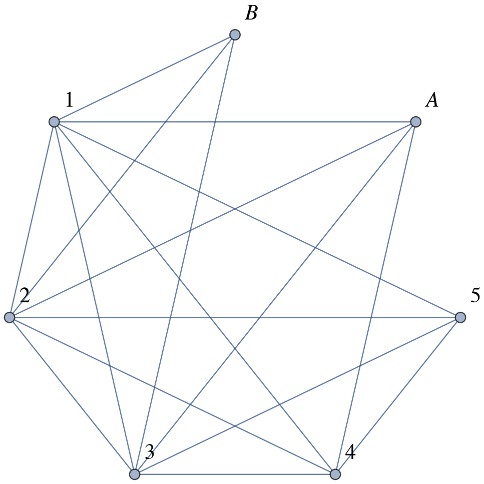
```
Binomial[7, 4]
```

```
35
```

However, if the graph is not a complete graph, it is not so easy to determine how many complete subgraphs of a certain size it contains. We turn to this problem next.

Consider the graph g_2 previously defined.

```
g2
```



Suppose we want to know how many K_4 subgraphs g_2 contains. We start with the list of its vertices.

```
V2 = VertexList[g2]
```

```
{1, 2, 3, 4, 5, A, B}
```

Next, we form the set of all subsets of size four of this set.

```
S2 = Subsets[V2, {4}]
```

```
{{1, 2, 3, 4}, {1, 2, 3, 5}, {1, 2, 3, A},
 {1, 2, 3, B}, {1, 2, 4, 5}, {1, 2, 4, A}, {1, 2, 4, B},
 {1, 2, 5, A}, {1, 2, 5, B}, {1, 2, A, B}, {1, 3, 4, 5},
 {1, 3, 4, A}, {1, 3, 4, B}, {1, 3, 5, A}, {1, 3, 5, B},
 {1, 3, A, B}, {1, 4, 5, A}, {1, 4, 5, B}, {1, 4, A, B},
 {1, 5, A, B}, {2, 3, 4, 5}, {2, 3, 4, A}, {2, 3, 4, B},
 {2, 3, 5, A}, {2, 3, 5, B}, {2, 3, A, B}, {2, 4, 5, A},
 {2, 4, 5, B}, {2, 4, A, B}, {2, 5, A, B}, {3, 4, 5, A},
 {3, 4, 5, B}, {3, 4, A, B}, {3, 5, A, B}, {4, 5, A, B}}
```

Here is an example of a subset of vertices that generates a complete graph in g_2.

```
CompleteGraphQ[g2, {1, 2, 3, 5}]
```

```
True
```

We now wish to repeat this calculation for all the subsets of size four.

```
tf = Table[CompleteGraphQ[g2, S2[[i]]], {i, Length[S2]}]
```

```
{True, True, True, True, True, True, False, False, False,
 False, True, True, False, False, False, False, False, False,
 False, False, True, True, False, False, False, False, False,
 False, False, False, False, False, False, False, False}
```

Finally, we count the number of times `True` occurs in `tf`.

```
Count[tf, True]
```

```
10
```

Let $\kappa(g, r)$ be the number of complete subgraphs with r vertices contained in the graph g. The following program implements $\kappa(g, r)$.

```
CountCompleteSubgraphs[g_, r_] :=
 Count[CompleteGraphQ[g, #] & /@ Subsets[VertexList[g], {r}],
  True]
```

Let us count the number of times K_3, K_4 and K_5 occur in the graph g_2; that is, we calculate $\kappa(g, 3)$, $\kappa(g, 4)$ and $\kappa(g, 5)$.

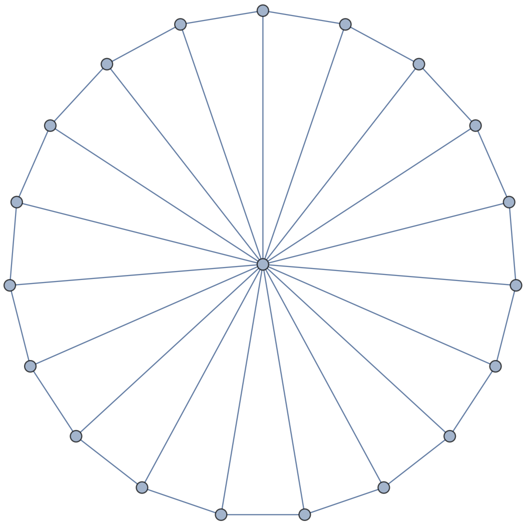
```
CountCompleteSubgraphs[g2, #] & /@ {3, 4, 5}
```

```
{19, 10, 2}
```

□ Bounding the Number of Complete Subgraphs of a Graph

Suppose we know that a certain graph g satisfies $\kappa(g, 3) \leq 19$. Can we determine the maximum of $\kappa(g, 4)$? Surely, g can have no K_4 subgraphs; that is, $\kappa(g, 4) = 0$, as in this example.

```
WheelGraph[20]
```



However, the example g_2 has been shown to have 19 K_3 subgraphs and 10 K_4 subgraphs.

In fact, we have shown in a joint paper [1] that for all graphs g with $\kappa(g, 3) \leq 19$, the maximum value of $\kappa(g, 4)$ is 10.

The following definition, due to Bollobás [2], is useful in what follows.

Definition

If $2 \leq p \leq r$, then $k_r(k_p \leq x)$ is the maximum number of K_r subgraphs that a graph can have if the number of its K_p subgraphs is less than or equal to x.

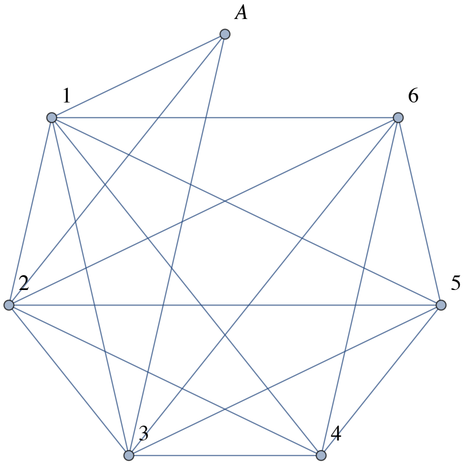
Thus, using this notation, we have shown in [1] that $k_4(k_3 \leq 19) = 10$.

A Closely Related Problem

Suppose now that the number of triangles a graph can have is fixed and we want to determine the graph with the fewest edges that can have that many triangles.

For example, suppose we want a graph having 23 triangles with the fewest possible edges. Our intuition is to look for graphs that are "tightly packed," that is, as close to complete graphs as possible. K_6 has 20 triangles and K_7 has 35. So let us start by adding a vertex A to K_6 and then add three edges from A to three of the vertices of K_6. This adds $\binom{3}{2} = 3$ new triangles.

```
h =
 EdgeAdd[
  VertexAdd[CompleteGraph[6, VertexLabels → Automatic], A],
  {A ⟷ 1, A ⟷ 2, A ⟷ 3}]
```



```
CountCompleteSubgraphs[h, 3]
```

```
23
```

```
CountCompleteSubgraphs[h, 2]
```

```
18
```

That shows that a graph with 23 triangles can be gotten with just 18 edges. Is there a graph with fewer edges that has 23 K_3 subgraphs?

The next theorem is due to Erdös and Hanani (see [3]).

Theorem

If a graph has x edges, where $x = \binom{s}{2} + \binom{t}{1}$ *and s is chosen as large as possible, then* $k_r(k_2 \le x) = \binom{s}{r} + \binom{t}{r-1}$.

If the number of edges is $x = 18$, then since $18 = \binom{6}{2} + \binom{3}{1}$, the theorem says that $k_3(k_2 \le 18) = \binom{6}{3} + \binom{3}{2} = 23$; that is, the maximum number of K_3 subgraphs in a graph with 18 edges is 23.

Is there a graph with fewer edges and 23 triangles? To answer this question, we can use the Erdös–Hanani theorem to compute the various maximum numbers of triangles with fewer edges. The program `ErdösHanani` gives the maximum number of triangles for a given number of edges, as determined by the Erdös–Hanani theorem; the table computes these values for edge numbers between three and 18.

```
ErdösHanani[x_] := Module[
  {s = 2, t = 1, x1},
  While[Binomial[s, 2] ⩽ x, s++];
  s = s - 1;
  x1 = x - Binomial[s, 2];
  If[x1 == 0, t = 0, t = x1];
  Binomial[s, 3] + Binomial[t, 2]
 ]
```

number of edges	maximum number of triangles
3	1
4	1
5	2
6	4
7	4
8	5
9	7
10	10
11	10
12	11
13	13
14	16
15	20
16	20
17	21
18	23

We see from the table that there is no graph with fewer than 18 edges having 23 triangles. Hence the fewest edges needed to produce a graph with exactly 23 triangles is indeed 18.

The Kruskal–Katona Bound

If we specify the number of triangles rather than the number of edges a graph can have, computing the maximum numbers of larger complete graphs is not as simple as in the previous section. Exact maximum numbers are not known in most cases, only upper bounds.

A well-known theorem of extremal graph theory (proved independently by Kruskal [4] and Katona [5]) can provide an upper bound for $\kappa(g, r)$, given $\kappa(g, p)$, $p < r$. In fact, the Kruskal–Katona result is for more general objects than graphs, but we will only be using it for graphs and only when $p = 3$; that is, we specify how many triangles a graph g can have and want to bound $\kappa(g, r)$ for some $r > 3$.

Theorem

Suppose a graph g has x triangles, where $x = \binom{s}{3} + \binom{t}{2} + \binom{u}{1}$, *where each of s, t, u is chosen in order and to be as large as possible at the time of choosing. Then for* $r > 3$, $k_r(k_3 \le x) \le \binom{s}{r} + \binom{t}{r-1} + \binom{u}{r-2}$.

For example, if the number of triangular subgraphs of g is $19 = \binom{5}{3} + \binom{4}{2} + \binom{3}{1}$, then the Kruskal–Katona upper bounds for $\kappa(g, 4)$ and $\kappa(g, 5)$ are $\binom{5}{4} + \binom{4}{3} + \binom{3}{2} = 12$ and $\binom{5}{5} + \binom{4}{4} + \binom{3}{3} = 3$.

To use this theorem in Mathematica, we first need to express x as the binomial sum, $x = \binom{s}{3} + \binom{t}{2} + \binom{u}{1}$. Given x, the function KK3 finds the numbers s, t, u.

```
KK3[x_] := Module[
  {s = 3, t = 2, u = 1, x1},
  While[Binomial[s, 3] ⩽ x, s++];
  s--;
  x1 = x - Binomial[s, 3];
  If[
   x1 == 0,
   t = 1,
   While[Binomial[t, 2] ⩽ x1, t = t + 1]
  ];
  t--;
  u = x1 - Binomial[t, 2];
  {s, t, u}
 ]
```

```
KK3[19]
```

```
{5, 4, 3}
```

Next, we define the functions K4K3 and K5K3, the Kruskal–Katona upper bounds for $\kappa(g, 4)$ and $\kappa(g, 5)$, given that $\kappa(G, 3) = x$.

```
K4K3[x_] := Module[
  {s, t, u},
  {s, t, u} = KK3[x];
  Binomial[s, 4] + Binomial[t, 3] + Binomial[u, 2]
 ]
```

```
K5K3[x_] := Module[
  {s, t, u},
  {s, t, u} = KK3[x];
  Binomial[s, 5] + Binomial[t, 4] + Binomial[u, 3]
 ]
```

Here are the results for 19 triangles.

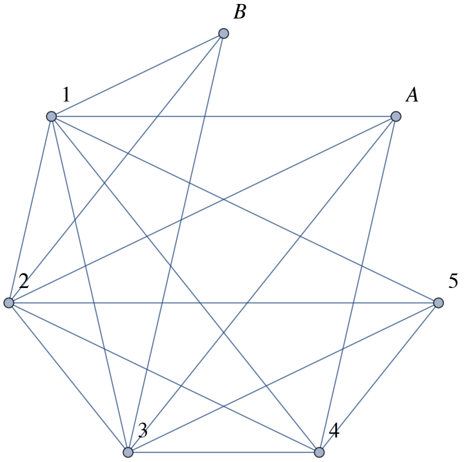
```
{K4K3[19], K5K3[19]}
```

```
{12, 3}
```

Thus, $k_4(k_3 \leq 19) \leq 12$ and $k_5(k_3 \leq 19) \leq 3$.

Here again is the graph g_2 with 19 triangles. It has 10 K_4 subgraphs and two K_5 subgraphs.

```
g2
```



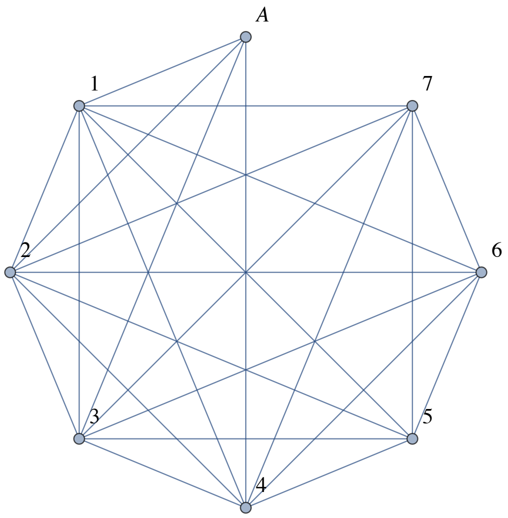
```
CountCompleteSubgraphs[g2, #] & /@ {3, 4, 5}
```

```
{19, 10, 2}
```

As mentioned, we proved in [1] that $k_4(k_3 \leq 19) = 10$ and $k_5(k_3 \leq 19) = 2$.

Finding $k_r(k_3 \leq x)$ exactly is often very difficult and results are not known in most cases. However, if there is a graph g with x triangles, $x = \binom{s}{3} + \binom{t}{2} + \binom{u}{1}$ with $\kappa(g, r) = \binom{s}{r} + \binom{t}{r-1} + \binom{u}{r-2}$ (the Kruskal–Katona bound), then $k_r(k_3 \leq x) = \kappa(g, r)$. Complete graphs are obvious examples. Also, as remarked in [2], if the number of triangles $x = \binom{s}{3} + \binom{t}{2}$, we define the graph h by adding to K_s a single vertex A and t edges joining A to t vertices of K_s; then $k_r(k_3 \leq x) = \kappa(h, r) = \binom{s}{r} + \binom{t}{r-1}$ (the Kruskal–Katona bound). For example, suppose $x = \binom{7}{3} + \binom{4}{2} = 41$. We construct such a graph, h_1.

```
h1 =
 EdgeAdd[
  VertexAdd[CompleteGraph[7, VertexLabels → Automatic], A],
  {A ⟷ 1, A ⟷ 2, A ⟷ 3, A ⟷ 4}]
```



```
CountCompleteSubgraphs[h1, #] & /@ {3, 4, 5}
```

```
{41, 39, 22}
```

```
{K4K3[41], K5K3[41]}
```

```
{39, 22}
```

Thus, $k_4(k_3 \leq 41) = 39$ and $k_5(k_3 \leq 41) = 22$.

Next, we find those numbers of triangles for which this construction works; that is, the third entry in their binomial representation is 0.

```
thirdEntryZero[x_] := Select[Range[x], KK3[#][[3]] == 0 &]
```

```
thirdEntryZero[100]
```

```
{1, 2, 4, 5, 7, 10, 11, 13, 16, 20, 21, 23, 26, 30, 35, 36, 38,
 41, 45, 50, 56, 57, 59, 62, 66, 71, 77, 84, 85, 87, 90, 94, 99}
```

Also, it is not difficult to see that if $x = \binom{s}{3} + \binom{t}{2} + \binom{1}{1}$, the Kruskal–Katona bound is the same as for $\binom{s}{3} + \binom{t}{2}$ (since $\binom{1}{w} = 0$ for $w > 1$). So the list of the numbers of triangles x for which $k_4(k_3 \leq x)$ is known can be expanded. Here are the known values for the first 100 positive integers.

```
known = Union[thirdEntryZero[100], thirdEntryZero[100] + 1]
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 16, 17, 20, 21,
 22, 23, 24, 26, 27, 30, 31, 35, 36, 37, 38, 39, 41, 42,
 45, 46, 50, 51, 56, 57, 58, 59, 60, 62, 63, 66, 67, 71,
 72, 77, 78, 84, 85, 86, 87, 88, 90, 91, 94, 95, 99, 100}
```

This leaves the following numbers of triangles up to 100 still unknown.

```
unknown = Complement[Range[100], known]
```

```
{9, 15, 18, 19, 25, 28, 29, 32, 33, 34, 40, 43, 44,
 47, 48, 49, 52, 53, 54, 55, 61, 64, 65, 68, 69, 70, 73,
 74, 75, 76, 79, 80, 81, 82, 83, 89, 92, 93, 96, 97, 98}
```

We have in fact settled (see [1]) the cases $x = 9, 15, 18, 19$ (where $k_4(k_3 \le x) = 2, 6, 9, 10$, respectively). We add these cases to the list of the known values.

```
known' = Union[known, {9, 15, 18, 19}]
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
 19, 20, 21, 22, 23, 24, 26, 27, 30, 31, 35, 36, 37, 38, 39,
 41, 42, 45, 46, 50, 51, 56, 57, 58, 59, 60, 62, 63, 66, 67,
 71, 72, 77, 78, 84, 85, 86, 87, 88, 90, 91, 94, 95, 99, 100}
```

And these are the unknown values of $k_4(k_3 \le x)$ for $x \le 100$.

```
unknown' = Complement[Range[100], known']
```

```
{25, 28, 29, 32, 33, 34, 40, 43, 44, 47, 48,
 49, 52, 53, 54, 55, 61, 64, 65, 68, 69, 70, 73, 74,
 75, 76, 79, 80, 81, 82, 83, 89, 92, 93, 96, 97, 98}
```

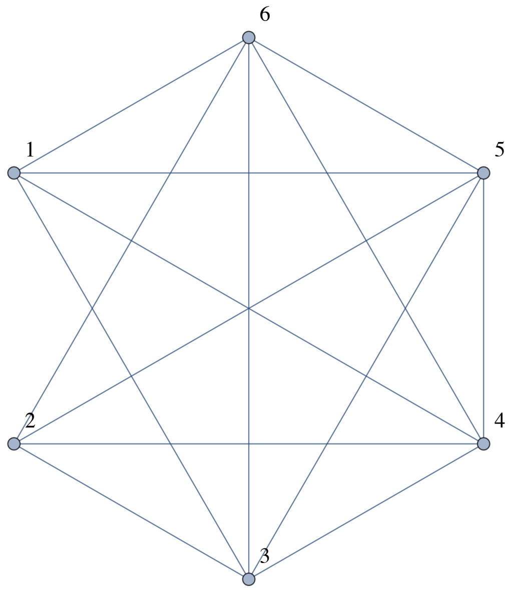
We had started listing the integer sequence $a_i = k_4(k_3 \le i)$ (see [6]), and the preceding results can be used to add to this sequence; for example, the first term of the sequence, $a_4 = 1$ (when there are four triangles allowed, the graph has at most one K_4); the sequence continued up to $a_{20} = 15$. Since we now know the consecutive values up to a_{24}, we can add four more consecutive values: $a_{21} = 15$, $a_{22} = 15$, $a_{23} = 16$, $a_{24} = 16$. Our conjecture in the next section implies that $a_{25} = 16$. Other selected values of the sequence can obviously also be obtained, given that we know so many of the first 100 cases.

Almost Complete Graphs and Turan Graphs

We have used complete graphs in building the maximal examples. However, even if we remove an edge from a complete graph, the number of K_4 subgraphs in this new graph `gr1` is the same as the Kruskal–Katona bound based on the number of K_3 subgraphs (calculated by `K4K3[gr1]`), as the following computation shows. (This can also be established with a simple argument that we omit.)

```
gr1[k_] :=
 EdgeDelete[CompleteGraph[k, VertexLabels → Automatic],
  {1 ⟷ 2}]
```

```
gr1[6]
```

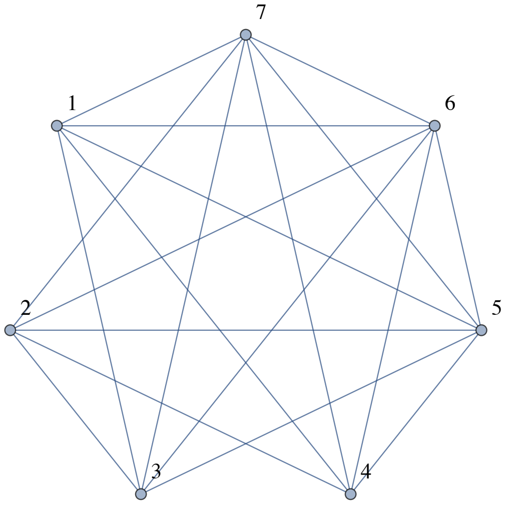


k	number of K_4	K4K3
	subgraphs	bound
5	2	2
6	9	9
7	25	25
8	55	55
9	105	105
10	182	182
11	294	294
12	450	450
13	660	660
14	935	935
15	1287	1287

In fact, even if we remove several edges from a complete graph, as long as they share a common vertex, the number of K_4 subgraphs in the resulting graph g and the Kruskal–Katona bound (based on the number `K4K3[g]` of K_3 subgraphs) remain the same, as the reader can easily check with Mathematica. However, if we remove two edges that do not share a common vertex, this is no longer the case, as the following computations show.

```
DeleteTwoEdges[k_] :=
 EdgeDelete[CompleteGraph[k, VertexLabels → Automatic],
  {1 ⟷ 2, 3 ⟷ 4}]
```

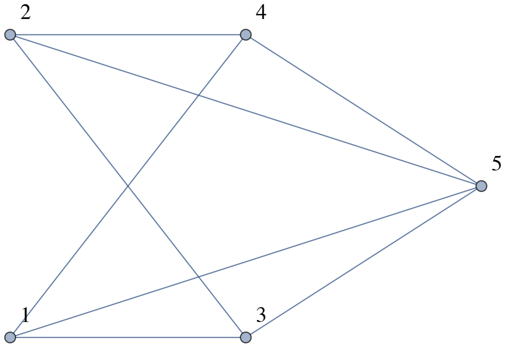
```
DeleteTwoEdges[7]
```



k	K_3 subgraphs	K_4 subgraphs	`K4K3` bound
6	12	4	5
7	25	16	17
8	44	41	42
9	70	85	86
10	104	155	156
11	147	259	260
12	200	406	407
13	264	606	607
14	340	870	871
15	429	1210	1211

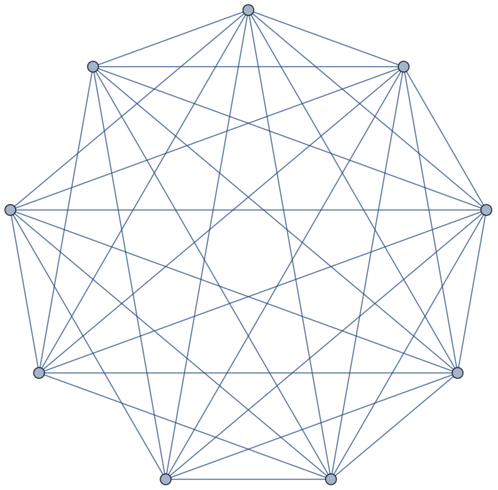
This time, the number of K_4 subgraphs is one less than the `K4K3` upper bound! These graphs are also Turan graphs. The Turan graph $T(n, k)$ is the graph formed by partitioning a set of n vertices into k subsets with sizes as equal as possible (differing by at most 1) and connecting two vertices by an edge if and only if they belong to different sets of the partition. The built-in Mathematica function `TuranGraph[n, k]` draws this graph. For example, $T(n, k)$ partitions the vertices $\{1, 2, 3, 4, 5\}$ into the subsets $\{1, 2\}$, $\{3, 4\}$, $\{5\}$; the edge `1 ⟷ 2` is omitted since it would connect vertices in the same subset, $\{1, 2\}$; `3 ⟷ 4` is omitted since it would connect vertices in the same subset $\{3, 4\}$.

```
TuranGraph[5, 3, VertexLabels → Automatic]
```



The graphs $T(k, k-2)$ are especially interesting to us as they are the same as the graphs `DeleteTwoEdges[k]`, defined before. It may not be immediately apparent that `TuranGraph[9, 7]` is the same graph as `DeleteTwoEdges[9]`; however, if we chose the right `GraphLayout` for the Turan graph, it becomes rather obvious.

```
TuranGraph[9, 7, GraphLayout → "CircularEmbedding"]
```



In addition, `IsomorphicGraphQ` can always be used to check.

```
IsomorphicGraphQ[DeleteTwoEdges[9], TuranGraph[9, 7]]
```

```
True
```

Therefore, for the Turan graph $T(k, k-2)$, the actual number of K_4 subgraphs is just one less than the Kruskal–Katona bounds! Do these graphs provide the true maximum values of K_4 subgraphs based upon the number of their triangles? We conjecture below that they do for $k > 6$. In addition, it is not hard to show that the number of triangles in $T(k, k-2)$ is

$4\,(k-4)+4\binom{k-4}{2}+\binom{k-4}{3}$. (This follows because the vertex sets of $T(k,k-2)$ are $\{1,2\}$, $\{3,4\}$, $\{5\}$, $\{6\}$, …, $\{k\}$; then consider all possible ways of choosing three elements from these sets without choosing two elements from the same set.) This expression can be simplified.

```
Factor[4 (k - 4) + 4 Binomial[k - 4, 2] + Binomial[k - 4, 3]]
```

```
1
─ (-4 + k) (-2 + k) (3 + k)
6
```

The sequence of these triangle numbers for the graphs $T(k,k-2)$ is the same (with an offset) as the sequence A000297 in [7]. (We recently added a comment to this entry that mentions this.)

Conjecture

If $k>6$, the Turan graph $T(k,k-2)$ has the maximum number of K_4 subgraphs for any graph with $1/6\,(k+3)\,(k-2)\,(k-4)$ triangles.

Finding Maximal Examples

Mathematica has a large database of graphs accessible with the function `GraphData`.

We wish to find maximal examples, that is, graphs that have the greatest number of K_4 subgraphs for their number of triangles. We make this precise with the following definition. A graph g is a K_4-maximal graph with respect to K_3 subgraphs if it has the greatest number of K_4 subgraphs for any graph with the same number of K_3 subgraphs (triangles) as g. We believe that these K_4-maximal graphs are “tightly packed” and thus have a relatively small number of vertices given their number of triangles. Suppose first we wish to find all the K_4-maximal graphs with exactly nine triangles. The Kruskal–Katona bound is three.

```
K4K3[9]
```

```
3
```

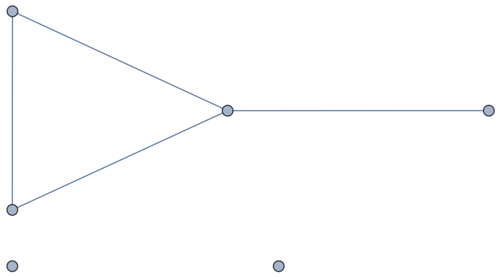
However, the maximal number of K_4 subgraphs is really two, that is, $k_4(k_3 \leq 9) = 2$ (see [1]). We first look for examples in the set of nonisomorphic graphs with six vertices, that is, subgraphs of K_6.

```
GraphData[6]
```

```
{{6, 12}, {6, 13}, {6, 14}, {6, 16}, {6, 19}, {6, 21}, {6, 24},
 {6, 25}, {6, 28}, {6, 29}, {6, 32}, {6, 33}, {6, 39}, {6, 41},
 {6, 42}, {6, 44}, {6, 45}, {6, 49}, {6, 51}, {6, 53}, {6, 55},
 {6, 57}, {6, 59}, {6, 60}, {6, 61}, {6, 65}, {6, 66},
 {6, 67}, {6, 69}, {6, 70}, {6, 71}, {6, 72}, {6, 73},
 {6, 74}, {6, 75}, {6, 76}, {6, 77}, {6, 78}, {6, 79},
 {6, 80}, {6, 81}, {6, 82}, {6, 83}, {6, 84}, {6, 86},
 {6, 88}, {6, 90}, {6, 92}, {6, 95}, {6, 96}, {6, 97},
 {6, 99}, {6, 100}, {6, 101}, {6, 102}, {6, 103}, {6, 104},
 {6, 105}, {6, 106}, {6, 108}, {6, 109}, {6, 110}, {6, 111},
 {6, 112}, {6, 114}, {6, 120}, {6, 122}, {6, 123}, {6, 124},
 {6, 125}, {6, 127}, {6, 128}, {6, 129}, {6, 130}, {6, 133},
 {6, 135}, {6, 136}, {6, 137}, {6, 138}, {6, 139}, {6, 140},
 {6, 141}, {6, 144}, {6, 148}, {6, 149}, {6, 150}, 2C3,
 2P2+2K1, 2P2+K1,3, 2P3, AGraph, AntennaGraph, {Barbell, 3},
 {BiggestLittlePolygon, 6}, {BlackBishop, {3, 4}}, C3+3K1,
 C3+P2+K1, C3+P3, C4+2K1, C4+P2, C5+K1, {Complete, 6},
 {CompleteBipartite, {2, 4}}, {CompleteTripartite, {1, 1, 4}},
 {CompleteTripartite, {1, 2, 3}}, {Cone, {3, 3}}, CrossGraph,
 {Cycle, 6}, DominoGraph, EGraph, {Empty, 6}, {Fan, {1, 5}},
 {Fan, {2, 4}}, FishGraph, {HamiltonLaceable, {6, 1}},
 {Hexahedral, 3}, {Hexahedral, 4}, {Hexahedral, 5},
 HGraph, K1,1,3+K1, K1,4+K1, K2,3+K1, K4+2K1,
 K4-e+2K1, K5+K1, {King, {2, 3}}, {LadderRung, 3},
 {Lollipop, {4, 2}}, {Lollipop, {5, 1}}, MothGraph,
 NetGraph, OctahedralGraph, P2+4K1, P2+K1,3, P2+K4,
 P2+K4-e, P3+3K1, P3+P2+K1, P4+2K1, P4+P2, P5+K1,
 {Pan, 5}, {Path, 6}, {Polyiamond, {4, 1}}, {Prism, 3},
 {Queen, {2, 3}}, RGraph, {Star, 6}, {Tadpole, {3, 3}},
 {Tadpole, {4, 2}}, {Tree, {6, 2}}, {TriangularGrid, 2},
 {Turan, {6, 5}}, UtilityGraph, W5+K1, {Wheel, 6}}
```

Here is the first such graph.

```
GraphData@{6, 12}
```

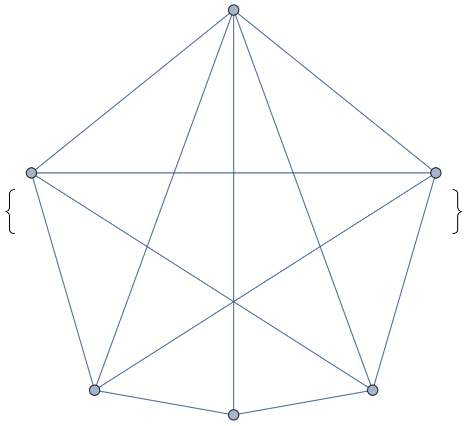


We apply `GraphData` to each entry to get the graphs themselves but suppress the large output.

```
graphdata6 = GraphData /@ GraphData[6];
```

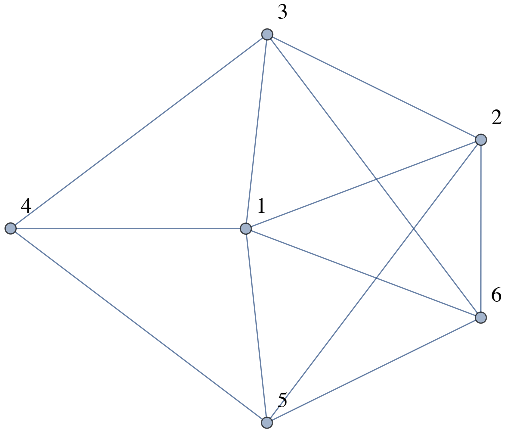
Within those, we next search for graphs with nine triangular subgraphs. There is only one.

```
triangles[6, 9] =
 Select[graphdata6, CountCompleteSubgraphs[#, 3] == 9 &]
```



We attach labels to the vertices of this graph for later use.

```
gr2 = Graph[EdgeList[First@triangles[6, 9]],
  VertexLabels → Automatic]
```



This graph `gr2` has two K_4 subgraphs.

```
CountCompleteSubgraphs[gr2, 4]
```

```
2
```

Hence `gr2` is a K_4-maximal example (see [1]).

This graph embedding is also extremal in another way; see [8].

We search for examples in the set of graphs with seven vertices, using the command `GraphData[7]`, which lists all non-isomorphic graphs with seven vertices. (If $k > 7$, `GraphData[k]` only lists some of the graphs with k vertices). `GraphData[7]` has 1044 entries, so the result is not immediate.

```
graphdata7 = GraphData /@ GraphData[7];
```

There are 35 examples of graphs with seven vertices and nine triangular subgraphs.

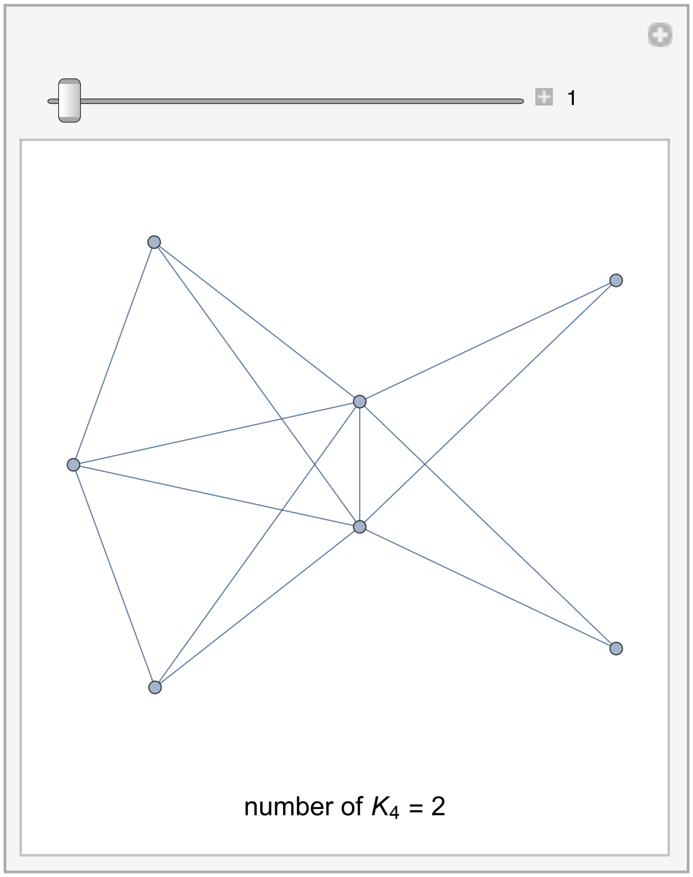
```
triangles[7, 9] =
  Select[graphdata7, CountCompleteSubgraphs[#, 3] == 9 &];
```

```
Length[%]
```

```
35
```

To see the individual graphs, we use Mathematica's built-in function `Manipulate`, where, in addition to the graph, the number of K_4 subgraphs of the graph is listed.

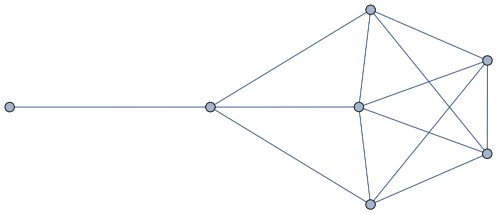
```
Manipulate[
 Text@
  Column[
   {Show[triangles[7, 9][[u]], ImageSize → {250, 250}],
    Spacer[20],
    Row[{"number of ", Style["K", Italic]_4, " = ",
      CountCompleteSubgraphs[triangles[7, 9][[u]], 4]}]
   }, Alignment → Center],
 {{u, 1, ""}, 1, 35, 1, Appearance → "Labeled"},
 SaveDefinitions → True]
```



number of K_4 = 2

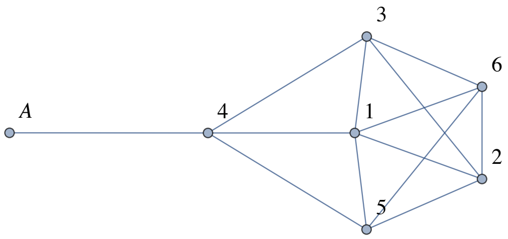
We suspect that one of the K_4-maximal graphs in `triangles[7, 9]` is really a simple modification of the single K_4-maximal graph with six vertices found in `GraphData[6]`; that is, the graph plus an edge. (We obviously do not want to add a triangle!) Stepping through the `Manipulate`, graph number 22 is easily seen to be that graph.

```
g3 = triangles[7, 9][[22]]
```



We then add the edge `A ⟷ 4` to the graph `gr2` found in `GraphData[6]` and finally, use `IsomorphicGraphQ` to verify that they are indeed isomorphic.

```
gr3 = EdgeAdd[VertexAdd[gr2, A], A ⟷ 4]
```

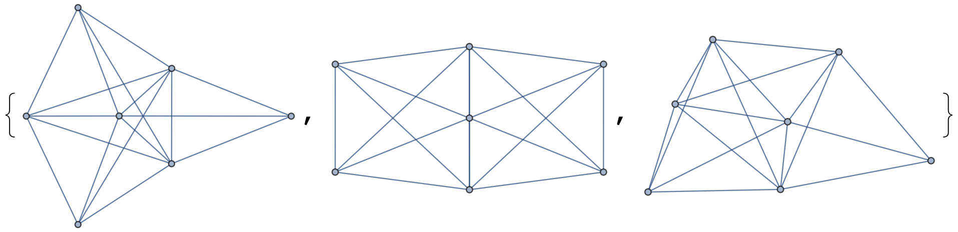


```
IsomorphicGraphQ[g3, gr3]
```

```
True
```

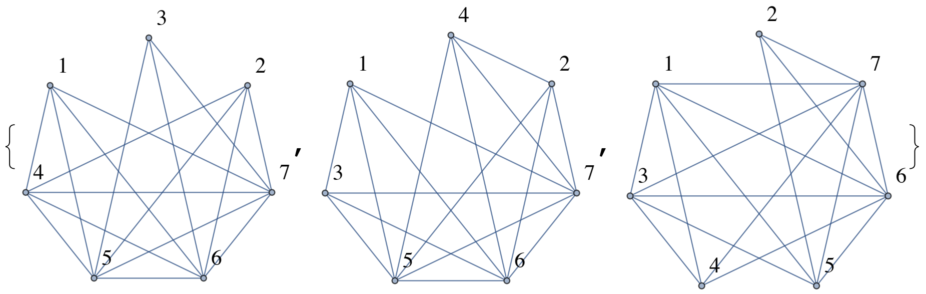
Next, suppose we want to find the K_4-maximal graphs with 19 triangles. Although K_6 has $\binom{6}{3} = 20$ triangles, if we remove even one edge from K_6, the resulting graph has only 16 triangles! Hence, there are no subgraphs of K_6 with exactly 19 triangles. However K_7 has $\binom{7}{3} = 35$ triangles; thus it is reasonable to search `GraphData[7]` for examples.

```
triangles[7, 19] =
 Select[graphdata7, CountCompleteSubgraphs[#, 3] == 19 &]
```



Sometimes there are hidden edges in graph drawings in Mathematica; the graph in the middle is a case in point (the edge from the center top to center bottom vertices cannot be seen). We therefore redraw the graphs, setting the option `GraphLayout`.

```
Graph[EdgeList[#], VertexLabels → "Name",
   GraphLayout → "CircularEmbedding"] & /@ triangles[7, 19]
```



We now ask for the number of K_4 subgraphs and the number of K_5 subgraphs in these graphs.

```
CountCompleteSubgraphs[#, 4] & /@ triangles[7, 19]
```

```
{10, 10, 10}
```

```
CountCompleteSubgraphs[#, 5] & /@ triangles[7, 19]
```

```
{2, 2, 2}
```

Also, all three graphs have the same number of edges, 17.

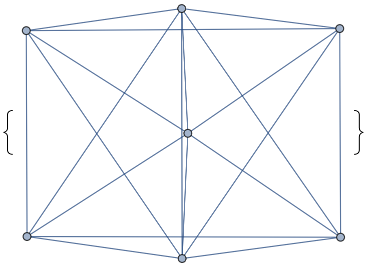
```
CountCompleteSubgraphs[#, 2] & /@ triangles[7, 19]
```

```
{17, 17, 17}
```

Thus, we have found three examples of graphs that illustrate our result of [1], that $k_4(k_3 \leq 19) = 10$.

We now ask what is the maximal number of K_4 subgraphs in a graph with 25 vertices. We again search for graphs with seven vertices and 25 triangles.

```
triangles[7, 25] =
 Select[graphdata7, CountCompleteSubgraphs[#, 3] == 25 &]
```



The one example we have found has 16 K_4 subgraphs, and the Kruskal–Katona bound for K_4 subgraphs is 17.

```
CountCompleteSubgraphs[triangles[7, 25][[1]], 4]
```

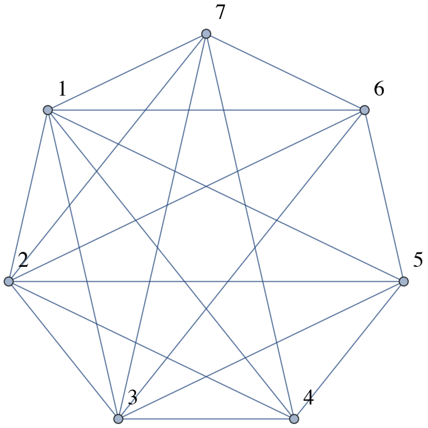
```
16
```

```
K4K3[25]
```

```
17
```

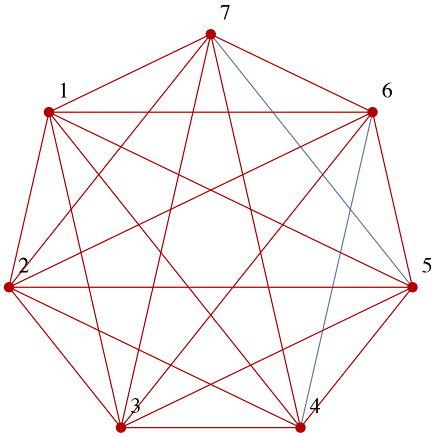
We investigate this graph further.

```
Graph[triangles[7, 25][[1]], VertexLabels → Automatic,
 GraphLayout → "CircularEmbedding"]
```



This is the Turan graph $T(7, 5)$; edges `4 ⧟ 6` and `5 ⧟ 7` are missing from K_7. This can also be seen with the `HighlightGraph` function.

```
HighlightGraph[CompleteGraph[7, VertexLabels → Automatic],
 triangles[7, 25][[1]]]
```



Since this is one of the Turan graphs and we have conjectured that they are K_4-maximal, we believe we have found a K_4-maximal example with 25 triangles.

Another example: Suppose we have a graph with 29 triangles. The Kruskal–Katona bound is 22.

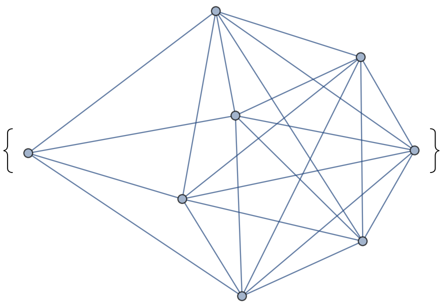
```
K4K3[29]
```

```
22
```

A search in `graphdata7` yields no graphs with 29 triangles. If we search in `GraphData[8]`, we find one graph with 29 triangles. It has 16 K_4 subgraphs.

```
graphdata8 = GraphData /@ GraphData[8];

triangles[8, 29] =
 Select[graphdata8, CountCompleteSubgraphs[#, 3] == 29 &]
```

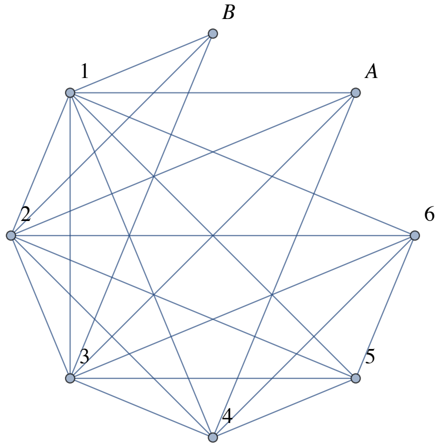


```
CountCompleteSubgraphs[triangles[8, 29][[1]], 4]
```

```
16
```

So it looks like the most K_4 subgraphs we can find for a graph with 29 triangles is 16. We can do better! For 26 triangles, since $26 = \binom{6}{3} + \binom{4}{2}$, the Kruskal–Katona bound is $\binom{6}{4} + \binom{4}{3} = 19$. Thus, if we add vertices A and B to K_6 and connect them to four and three vertices of K_6, respectively, we get a graph with 29 triangles and 20 K_4 subgraphs.

```
triangles[8, 29] =
 EdgeAdd[
  VertexAdd[CompleteGraph[6, VertexLabels → Automatic],
   {A, B}], {A ⟷ 1, A ⟷ 2, A ⟷ 3, A ⟷ 4, B ⟷ 1, B ⟷ 2, B ⟷ 3}]
```



```
{CountCompleteSubgraphs[triangles[8, 29], 3],
 CountCompleteSubgraphs[triangles[8, 29], 4]}
```

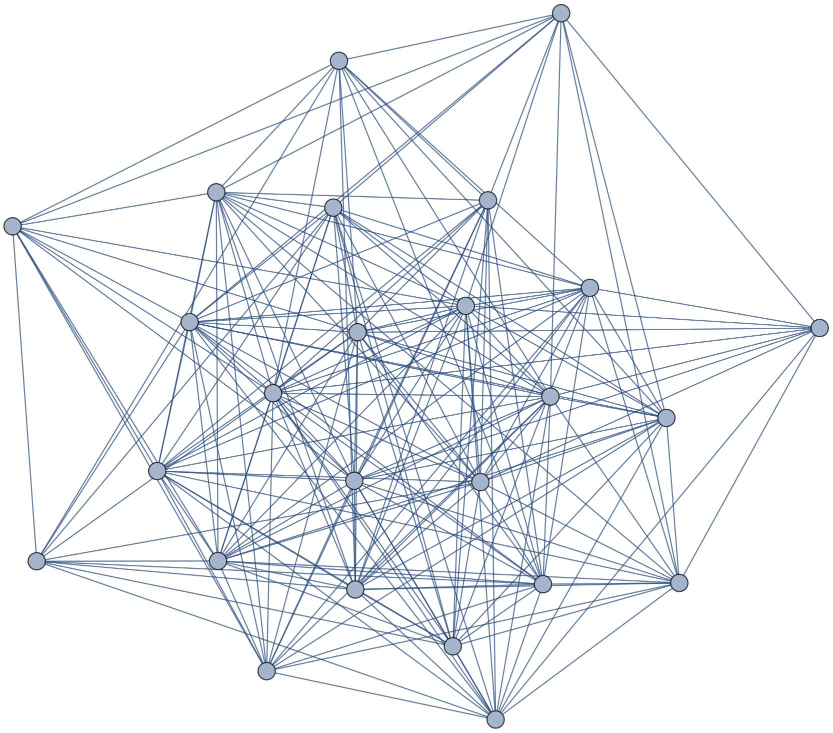
```
{29, 20}
```

The graph `triangles[8, 29]` has eight vertices, but it did not show up in `GraphData[8]`, which has only 289 out of 12346 possible graphs. `GraphData[7]` contains all 1044 graphs with seven vertices.

□ A Larger Example

We have looked at examples with relatively few vertices, and these examples might give the impression that the Kruskal–Katona bounds do not differ significantly from the real values—so why expend so much effort trying to improve them? We construct a larger example to show that the difference can be quite large.

```
g4a = RandomGraph[{25, 200}]
```



```
a4 = {CountCompleteSubgraphs[g4a, 3],
   CountCompleteSubgraphs[g4a, 4]}
```

```
{700, 1246}
```

Here is another example.

```
g4b = RandomGraph[{25, 200}];
b4 = {CountCompleteSubgraphs[g4b, 3],
   CountCompleteSubgraphs[g4b, 4]}
```

```
{666, 989}
```

The Kruskal–Katona bound, however, usually yields more than twice as many K_4 subgraphs! More extreme examples can easily be constructed.

```
K4K3@*First /@ {a4, b4}
```

```
{2410, 2275}
```

Conclusion

There have been some efforts to improve the Kruskal–Katona bounds in the case of graphs (see, for example [1] and [9]); however these have had very limited success. We feel that not enough insight into this problem has been gained and that perhaps by using computer experiments, conjectures can formulated and then proved to advance our knowledge in this area. For example, if we knew that a maximal K_4 example with 19 triangles must occur in a graph with seven vertices, our search of `GraphData[7]` would be sufficient to prove that the maximum number of K_4 subgraphs a graph with 19 K_3 subgraphs can have is 10. We succeeded in proving this result in [1] without using computers, but only with a great deal of effort.

To read more on using Mathematica’s graph theory capability to investigate other maximal problems in graph theory, see [10] and [11].

Acknowledgement

I wish to thank the editor, whose sage advice substantially improved this paper’s programing and content.

References

[1] R. Cowen and W. Emerson, “On Finding $k_4(k_3 \leq x)$,” *Graph Theory Notes, New York Academy of Sciences*, **34**, 1998 pp. 26–30.
www.researchgate.net/publication/287991696_On_Finding_k4k3_x.

[2] B. Bollobás, “Relations between Sets of Complete Subgraphs,” *Proceedings of the Fifth British Combinatorial Conference*, Aberdeen, 1975 pp. 79–84.
www.researchgate.net/publication/268543809_Relations_between_sets_of_complete_subgraphs.

[3] P. Erdös, “On the Number of Complete Subgraphs Contained in Certain Graphs,” *Publications of the Mathematics Institute of the Hungarian Academy of Sciences*, **7**, 1962 pp. 459–464.

[4] J. Kruskal, “The Number of Simplices in a Complex,” *Mathematical Optimization Techniques* (R. Bellman, ed.), Berkeley: University of California Press, 1963 pp. 251–278.

[5] G. Katona, “A Theorem of Finite Sets,” *The Theory of Graphs* (P. Erdös, ed.), Budapest: Akadémia Kiadó, 1968 pp. 187–207.

[6] N. J. A. Sloane. *The Online Encyclopedia of Integer Sequences*. oeis.org/A020917.

[7] N. J. A. Sloane. *The Online Encyclopedia of Integer Sequences*. oeis.org/A000297.

[8] E. W. Weisstein. "Graham's Biggest Little Hexagon" from MathWorld—A Wolfram Web Resource. mathworld.wolfram.com/GrahamsBiggestLittleHexagon.html.

[9] A. Frohmader, "A Kruskal–Katona Type Theorem for Graphs." arxiv.org/abs/0710.3960.

[10] E. W. Weisstein. "Cage Graph" from MathWorld—A Wolfram Web Resource. mathworld.wolfram.com/CageGraph.html.

[11] E. W. Weisstein. "Degree-Diameter Problem" from MathWorld—A Wolfram Web Resource. mathworld.wolfram.com/Degree-DiameterProblem.html.

R. Cowen, "Improving the Kruskal–Katona Bounds for Complete Subgraphs of a Graph," *The Mathematica Journal*, 2018. dx.doi.org/doi:10.3888/tmj.20-6.

About the Author

Robert Cowen is Professor Emeritus in Mathematics, Queens College, CUNY. He does research in logic, combinatorics and set theory. He taught a course in Mathematica programming for many years, emphasizing discovery of mathematics, and is currently working on a text on learning Mathematica through discovery with John Kennedy. His website is sites.google.com/site/robertcowen.

Robert Cowen
16422 75th Avenue
Fresh Meadows, NY 11366
robert.cowen@gmail.com