Pensieve header: October 20: Some Hochschild Homology.

Today. Trees from triangulations, then some Hochschild homology, then whatever you may suggest, then EIWL 9-12, then, if time, Patterns.

Topics (in no particular order). Whatever you may suggest; whatever comes to my mind; ~~the Fibonacci numbers~~; the Catalan numbers; ~~the Jones polynomial~~; ~~a more efficient Jones algorithm~~; ~~a riddle on spheres~~; Khovanov homology; Γ-calculus; the Hopf fibration; Hilbert's 13th problem; non-commutative Gaussian elimination; free Lie algebras; the Baker-Campbell-Hausdorff formula; wacky numbers; an order 4 torus; the Schwarz Lantern; knot colourings; the Temperley-Lieb pairing; the dodecahedral link; sound experiments; barycentric subdivisions; a Peano curve; braid closures and Vogel's algorithm; the insolubility of the quintic; phase portraits; the Mandelbrot set; shadows of the Cantor aerogel; quilt plots; some image transformations; De Bruijn graphs; the Riemann series theorem; finite type invariants and the Willerton fish; ~~the Towers of Hanoi~~; Hochschild homology of (some) coalgebras; convolutions and image improvements.

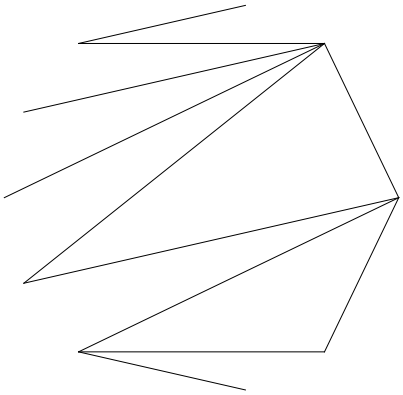
An Image Manipulation Challenge

The image at http://drorbn.net/bbs/show?shot=17-1750-171016-111042.jpg is pathetic. Can you improve it? Whatever you do, should also work well with all other images at http://drorbn.net/bbs/show.php?prefix=17-1750.

Trees from Triangulations

```
triang =
  ds[d[9, 11], d[9, 12], d[0, 12], d[0, 9], d[2, 7], d[2, 6], d[3, 5], d[2, 5], d[2, 8], d[0, 8], d[0, 2]];
```

```
triang /. ds[ls___] → Graphics[{ls}] /. d[i_, j_] :↦ Line[{i, j}] /. j_Integer :↦ {Cos[2 π j / 14], Sin[2 π j / 14]}
```



```
Last[d[0, 13] (Times @@ triang) ∏_{j=0}^{12} e[j, j + 1, ●] //.
   e[i_, j_, t1_] e[j_, k_, t2_] d[i_, k_] :↦ e[i, k, p[t1, t2]]] // TreeForm
```

Some Hochschild Homology

9. Interactive Manipulation

```
Manipulate[Table[Orange, n], {n, 1, 5, 1}]
```

```
Table[Table[Orange, n], {n, 1, 5, 1}]
```

```
Manipulate[Column[{n, n^2, n^3}], {n, 1, 10, 1}]
```

```
Table[Column[{n, n^2, n^3}], {n, 1, 10, 1}]
```

```
Manipulate[Column[{n, n^2, n^3}], {n, 1, 10}]
```

```
Manipulate[BarChart[{1, a, 4, 2 * a, 4, 3 * a, 1}], {a, 0, 5}]
```

```
Manipulate[PieChart[{1, a, 4, 2 * a, 4, 3 * a, 1}], {a, 0, 5}]
```

```
Manipulate[Graphics[Style[RegularPolygon[n], Hue[h]]], {n, 5, 20, 1}, {h, 0, 1}]
```

```
Manipulate[Graphics[Style[RegularPolygon[5], color]], {color, {Red, Yellow, Blue}}]
```

10. Images

```
CurrentImage[]
```

```
$ImagingDevices
```

```
$ImagingDevice = $ImagingDevices[[2]];
img = CurrentImage[]
```

```
ColorNegate[img]
```

```
{Blur[img], Blur[img, 10]}
```

```
Table[Blur[img, n], {n, 0, 15, 5}]
```

```
ImageCollage[Table[Blur[img, n], {n, 0, 15, 5}]]
```

```
DominantColors[img]
```

```
Binarize[img]
```

```
Manipulate[Binarize[img, t], {t, 0, 1}]
```

```
DominantColors[Binarize[img]]
```

```
img1 = EdgeDetect[img]
```

```
ImageAdd[img, img1]
```

```
imgs = WikipediaData["knot theory", "ImageList"]
```

```
ImageCollage[Scaled[1] → imgs, Method → "ClosestPacking", Background → White]
```

```
cf = Import["http://drorbn.net/ap/Classes/17-1750-ShamelessMathematica/20170929_110340.jpg"]
```

```
EdgeDetect[cf]
```

```
faces = FindFaces[cf]
```

```
ImageTrim[cf, #] & /@ faces
```

11. Strings and Text

```
"This is a string."
```

```
StringLength["hello"]
```

```
StringReverse["hello"]
```

```
ToUpperCase["I'm coding in the Wolfram Language!"]
```

```
StringTake["this is about strings", 10]
```

```
StringLength[StringTake["this is about strings", 10]]
```

```
StringJoin["Hello", " ", "there!", " How are you?"]
```

```
{"apple", "banana", "strawberry"}
```

```
StringTake[{"apple", "banana", "strawberry"}, 2]
```

```
StringJoin[{"apple", "banana", "strawberry"}]
```

```
Characters["a string is made of characters"]
```

```
Sort[Characters["a string of characters"]]
```

```
InputForm[Sort[Characters["a string of characters"]]]
```

```
TextWords["This is a sentence. Sentences are made of words."]
```

```
StringLength[TextWords["This is a sentence. Sentences are made of words."]]
```

```
StringTake[WikipediaData["knot theory"], 100]
```

```
WordCloud[DeleteStopwords[WikipediaData["knot theory"]]]
```

```
Take[WordList[], 20]
```

```
WordCloud[StringTake[WordList[], 1]]
```

```
RomanNumeral[1988]
```

```
Table[RomanNumeral[n], {n, 20}]
```

```
ListLinePlot[Table[StringLength[RomanNumeral[n]], {n, 100}]]
```

```
IntegerName[56]
```

```
ListLinePlot[Table[StringLength[IntegerName[n]], {n, 100}]]
```

```
Alphabet[]
```

```
LetterNumber[{"a", "b", "x", "y", "z"}]
```

```
FromLetterNumber[{10, 11, 12, 13, 14, 15}]
```

```
Alphabet["Russian"]
```

```
Rasterize[Style["ABC", 100]]
```

```
EdgeDetect[Rasterize[Style["ABC", 100]]]
```

```
FromCharacterCode /@ Range[1000]
```

12. Sound

```
Sound[SoundNote["C"]]
```

```
Sound[{SoundNote["C"], SoundNote["C"], SoundNote["G"]}]
```

```
Sound[Table[SoundNote[RandomInteger[12], 0.1, "Violin"], 20]]
```

```
Play[Sin[440 × 2 Pi t], {t, 0, 1}]
```

Patterns

One of the unique strengths of the Wolfram Language is its powerful and succinct—yet highly readable—symbolic pattern language. Convenient both for immediate use in individual functions, and for systematic large-scale programming, the Wolfram Language's pattern language generalizes concepts like regular expressions to describe general patterns for arbitrary symbolic structures.

Basic Pattern Objects

_ (**Blank**)— any expression (a "blank" to be filled in)

***x*_** — any expression, to be referred to as *x*

__ (**BlankSequence**)— any sequence of one or more expressions

___ (**BlankNullSequence**)— any sequence of zero or more expressions

Composite Patterns

p*|*p*|*p (**Alternatives**) — any of several alternatives

***p*..** (**Repeated**), ***p*...** (**RepeatedNull**) — a pattern to be repeated

x*:*p (**Pattern**) — an arbitrary pattern, to be referred to as *x*

Except — anything except a specified pattern

Longest, **Shortest** — longest, shortest possible matches

OptionsPattern ▪ **PatternSequence** ▪ **Verbatim** ▪ **HoldPattern**

OrderlessPatternSequence — elements in any order

KeyValuePattern — an association or list of rules containing specified elements

Restrictions on Patterns

_h — pattern with a specified head *h* (e.g. **_Integer**)

Condition (**/;**) — condition on a pattern (e.g. $x\_ \,/;\, x > 7$)

PatternTest (**?**) — pattern with a function test (e.g. **_?NumberQ**)

Pattern Defaults

_*:*e (**Optional**) — pattern that defaults to *e* if omitted

_. (**Optional**) — pattern with predefined default

Default — predefined default arguments for a function

Attributes Affecting Patterns

Orderless ▪ **Flat** ▪ **OneIdentity**
