Advent of Code 2023

Day 3

In[1]:=
```
SetDirectory[NotebookDirectory[]]
```

Out[1]= /home/rlupi/src/aoc2023/3

In[2]:=
```
testInput = StringSplit["467..114..
...*......
..35..633.
......#...
617*......
.....+.58.
..592.....
......755.
...$.*....
.664.598..", EndOfLine];
```

In[3]:=
```
input = Import["input.txt", "Lines"];
```

In[4]:=
```
Length[input]
```

Out[4]= 140

I will use unevaluated forms to mark digits in the input data. So define a custom formatting to make the display less cluttered:

In[5]:=
```
maybe /: MakeBoxes[maybe[n_], StandardForm] :=
  FrameBox@StyleBox[MakeBoxes[n, StandardForm], FontColor → Gray, FontWeight → Bold];
digit /: MakeBoxes[digit[n_], StandardForm] :=
  FrameBox@StyleBox[MakeBoxes[n, StandardForm], FontColor → Red, FontWeight → Bold];
schematic /: MakeBoxes[schematic[data_], StandardForm] :=
 MakeBoxes[MatrixForm[data], StandardForm]
```

In[8]:=
```
{digit[n], maybe[n]}
```

Out[8]= {n, n}

Part 1

This day was harder than I expected. I went off-track trying to use CellularAutomata, but it's too

complex.

In[9]:=
```
ParseSchematic[input_] := StringCases[input,
    {n : DigitCharacter :> maybe[Interpreter[Integer][n]],
     p : PunctuationCharacter :> p
    }] // schematic;
testSchematic = ParseSchematic[testInput]
```

Out[10]=

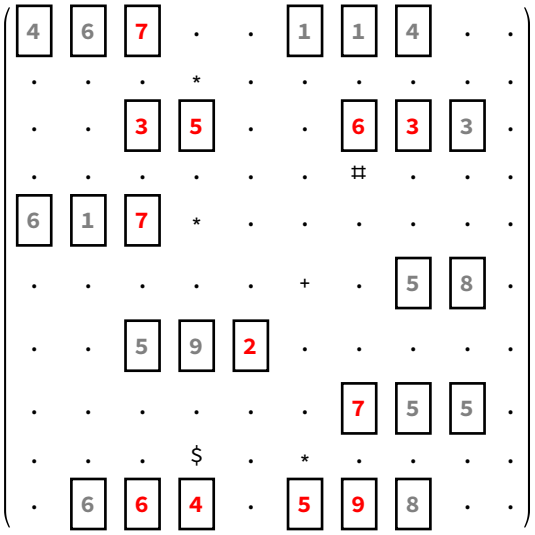
4	6	7	.	.	1	1	4	.	.
.	.	.	*	.	.	.	.	.	.
.	.	3	5	.	.	6	3	3	.
.	.	.	.	.	.	#	.	.	.
6	1	7	*	.	.	.	.	.	.
.	.	.	.	.	+	.	5	8	.
.	.	5	9	2	.	.	.	.	.
.	.	.	.	.	.	7	5	5	.
.	.	.	$	.	*	.	.	.	.
.	6	6	4	.	5	9	8	.	.

In[11]:=
```
ArraySweep[f_][a_] :=
  ArrayFilter[f[##][2, 2], DeleteCases[ArrayRules[##], {2, 2} → _] &, a, 1, Padding → "."];
```

In[12]:=
```
MarkDigits[schematic[sc_]] := Module[{mark},
   mark[maybe[d_],
     neighbours_ /; Count[neighbours, {_, _} → c_String /; c ≠ "."] > 0] := digit[d];
   mark[x_, _] := x;
   ArraySweep[mark][sc] // schematic
  ];
```

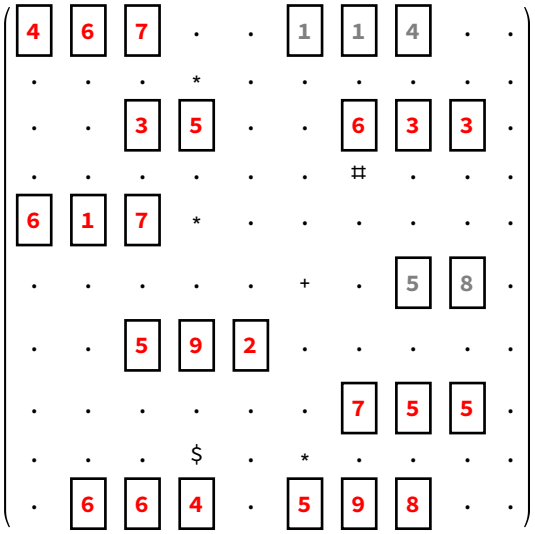
In[13]:=
```
MarkDigits[testSchematic]
```

Out[13]=



In[14]:=
```
ExpandDigits[schematic[sc_]] := Module[{expand},
  expand[maybe[d_], neighbours_ /; Count[neighbours, {_, _} → digit[_]] > 0] := digit[d];
  expand[x_, _] := x;
  FixedPoint[ArraySweep[expand], sc] // schematic
 ];
MarkDigits[testSchematic] // ExpandDigits
```

Out[15]=



In[16]:=
```
ExtractNumbers[schematic[m_]] := Block[{numbers},
  numbers = Map[SequenceCases[#, {digit[_] ..}] &, m];
  numbers = numbers /. {digit[d_] :> d, {} → Nothing};
  Apply[FromDigits[{##}] &, numbers, {2}] // Flatten
 ]
MarkDigits[testSchematic] // ExpandDigits // ExtractNumbers
```

Out[17]=
```
{467, 35, 633, 617, 592, 755, 664, 598}
```

In[18]:=
```
MarkDigits[testSchematic] // ExpandDigits // ExtractNumbers // Total
```

Out[18]=
```
4361
```

Ok, let's see it on real data:

In[19]:=
```
ParseSchematic[input] // MarkDigits // ExpandDigits // ExtractNumbers // Total
```

Out[19]=
```
540 212
```

Part 2

In[20]:=
```
part /: MakeBoxes[part[unique_, n_], StandardForm] := UnderscriptBox[
    FrameBox@StyleBox[MakeBoxes[n, StandardForm], FontWeight → Bold],
    MakeBoxes[unique, StandardForm]];
sym /: MakeBoxes[sym[unique_, class_], StandardForm] :=
  UnderscriptBox[MakeBoxes[class, StandardForm], MakeBoxes[unique, StandardForm]];
{part[Unique[], 345], sym[Unique[], "*"]}
```

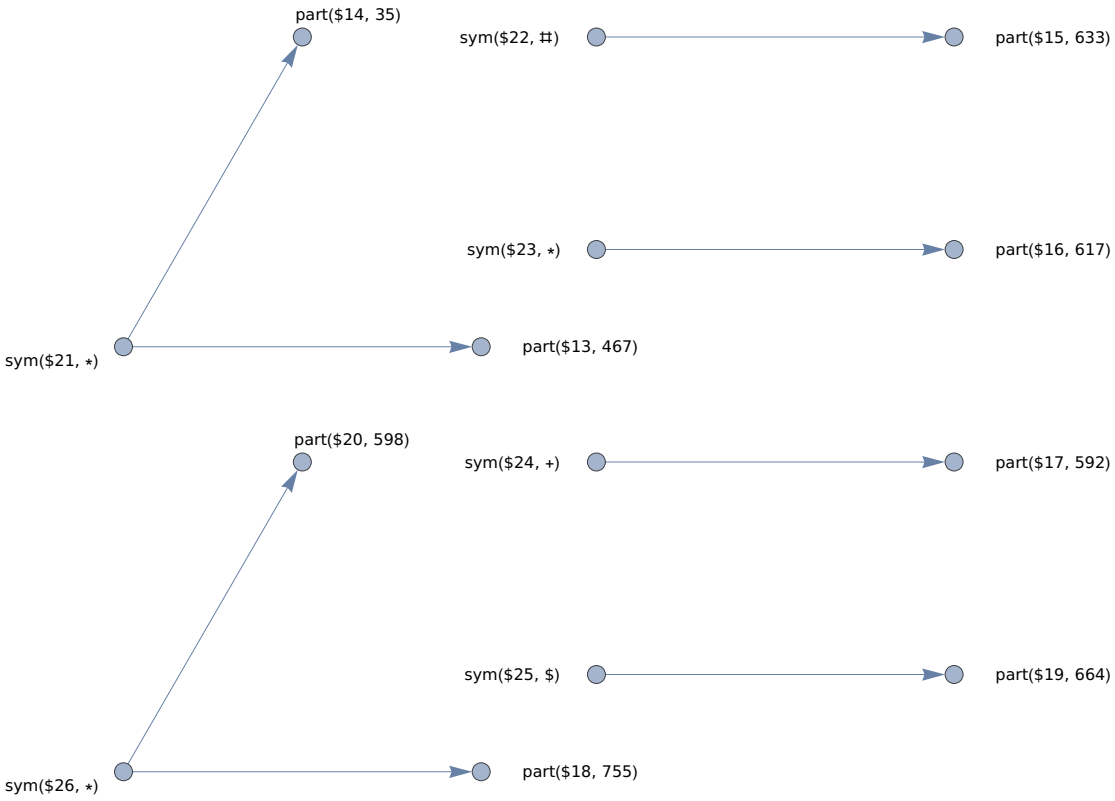
Out[22]=

$$\left\{\underset{\$11}{\boxed{345}},\ \underset{\$12}{*}\right\}$$

In[23]:=
```
EngineGraph[schematic[sc_]] := Block[{m, numbers, parts, connect, g},
  m = schematic[sc] // MarkDigits // ExpandDigits // First
  (* First just to strip schematic *);
  Echo[Head@m];
  parts = Map[SequenceReplace[#, p : {digit[_] ..} :>
      Module[{u = Unique[]},
       Splice@Table[part[u, FromDigits@Apply[Identity, p, {1}]], Length[p]]] &, m];
  parts = parts /. {maybe[_] -> ".", "." -> ".", c_String :> sym[Unique[], c]};
  Attributes[connect] = {Listable};
  connect[_, {_, _} -> "."] := Nothing;
  connect[".", _] := Nothing;
  connect[p_, _] := Nothing;
  connect[s_sym, {_, _} -> p_part] := s -> p;
  g = ArrayFilter[connect[#[[2, 2]], DeleteCases[ArrayRules[#], {2, 2} -> _]] &,
    parts, 1, Padding -> "."] // Flatten // DeleteDuplicates;
  Graph[g]
 ];
testEngine = Graph[EngineGraph[testSchematic],
  VertexLabels -> "Name", GraphLayout -> "PlanarEmbedding", ImageSize -> Large]
```

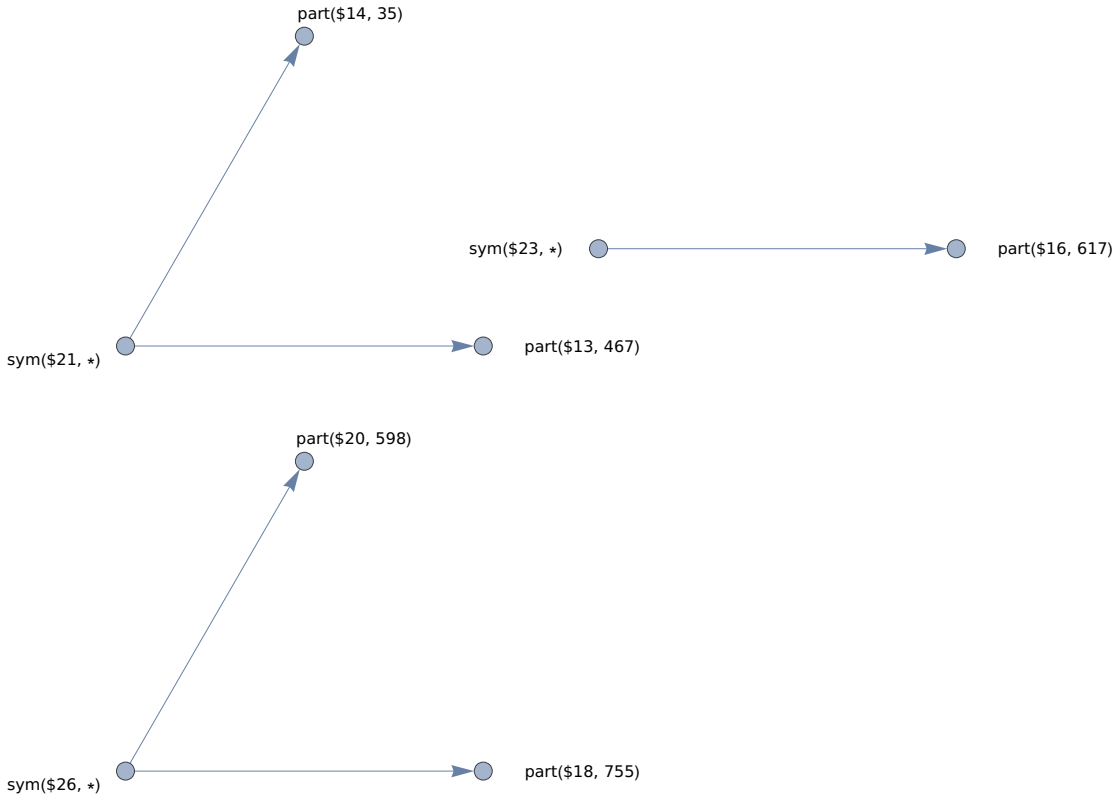
» List

Out[24]=



In[25]:=
```
NeighborhoodGraph[testEngine, sym[_, "*"], VertexLabels → "Name"]
```

Out[25]=



In[26]:=
```
AdjacencyList[%]
```

Out[26]=

$$\{\{\underset{\$13}{\boxed{467}}, \underset{\$14}{\boxed{35}}\}, \{\underset{\$16}{\boxed{617}}\}, \{\underset{\$18}{\boxed{755}}, \underset{\$20}{\boxed{598}}\}, \{\underset{\$21}{*}\}, \{\underset{\$21}{*}\}, \{\underset{\$23}{*}\}, \{\underset{\$26}{*}\}, \{\underset{\$26}{*}\}\}$$

In[27]:=
```
GearRatios[engine_] := Cases[
    NeighborhoodGraph[engine, sym[_, "*"]] // AdjacencyList,
    {part[_, a_], part[_, b_]} :→ a * b
  ];
GearRatios[testEngine] // Total
```

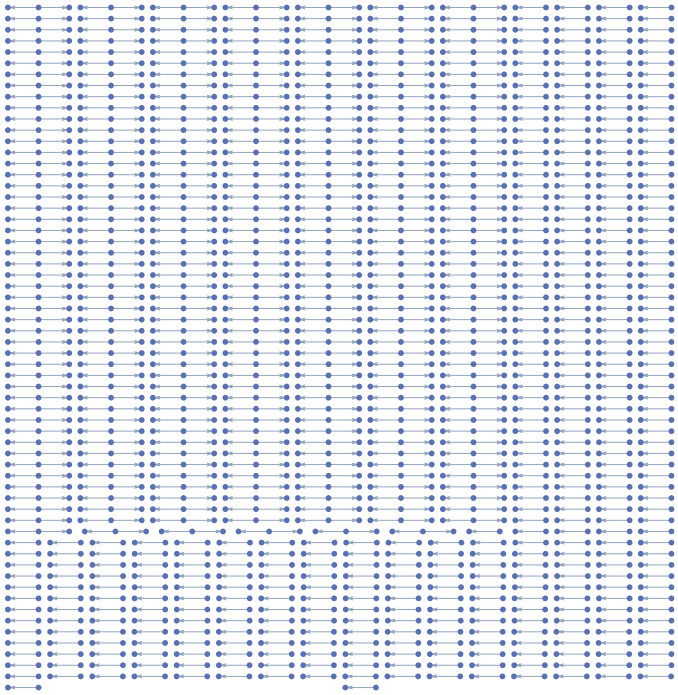
Out[28]=

467 835

In[29]:=
```
inputEngine = ParseSchematic[input] // EngineGraph
```

» List

Out[29]=



In[30]:=
```
inputEngine // GearRatios
```

Out[30]=

{114 573, 112 993, 153 216, 418 264, 103 368, 107 319, 379 324, 536 635, 14 592, 424 615, 507 848,
939 378, 237 600, 869 045, 141 382, 31 520, 389 298, 629 391, 46 980, 178 920, 55 242,
25 056, 635 034, 290 580, 137 280, 119 035, 358 479, 2205, 718 200, 92 400, 202 982,
42 828, 799, 23 999, 162 710, 53 836, 45 315, 246 402, 197 308, 487 200, 35 990, 986 049,
77 748, 107 648, 123 384, 344 915, 60 588, 425 130, 172 500, 356 310, 77 280, 5883, 4616,
324 919, 292 400, 85 600, 28 993, 2688, 697 375, 202 880, 296 010, 542 100, 346 500,
303 804, 1562, 462 429, 744 753, 8372, 136 940, 58 146, 864, 19 734, 108 052, 27 474,
16 192, 44 275, 226 754, 7294, 262 416, 536 300, 687 051, 165 483, 141 246, 323 136,
22 200, 309 072, 230 510, 266 420, 607 948, 238 720, 1209, 376 248, 65 817, 114 450,
33 110, 44 226, 301 760, 492 156, 295 974, 39 600, 186 258, 35 400, 234 150, 585 024,
194 691, 72 412, 326 915, 13 618, 511 875, 467 138, 791 700, 265 905, 165 240, 506 880,
323 818, 74 572, 131 077, 56 376, 77 760, 56 835, 335 685, 269 726, 497 140, 137 268,
117 572, 529 549, 407 790, 78 624, 46 336, 771 524, 325 464, 155 925, 31 200, 593 732,
239 280, 30 580, 263 980, 585 984, 70 252, 64 120, 695 273, 81 534, 163 280, 334 425,
268 593, 95 484, 52 438, 36 450, 136 644, 12 495, 321 685, 370 560, 348 234, 60 138,
69 048, 5264, 194 500, 132 132, 214 676, 930 150, 62 237, 591 414, 96 240, 121 360,
23 301, 140 990, 262 484, 27 930, 203 442, 98 154, 123 228, 385 008, 635 520, 59 109,
620 312, 94 658, 157 320, 95 106, 140 010, 215 418, 779 520, 82 181, 354 626, 359 788,
7920, 290 523, 2828, 390 648, 75 780, 117 199, 5652, 14 762, 533 540, 879 522, 248 325,
586 296, 344 470, 223 539, 272 720, 225 645, 155 511, 54 510, 99 981, 436 896, 44 255,
507 952, 725 696, 355 623, 893 730, 335 465, 41 904, 141 332, 105 984, 573 678, 267 804,
830 320, 37 696, 287 280, 543 361, 8944, 211 575, 722 638, 50 260, 168 516, 835 347,
321 763, 646 000, 853 432, 78 624, 669 390, 228 900, 327 651, 20 470, 258 390, 133 892,
163 014, 368 510, 244 800, 452 226, 266 250, 553 220, 42 616, 89 344, 520 047, 86 154,
767 457, 347 260, 268 640, 557 004, 270 720, 167 856, 805, 309 650, 417 720, 165 132,
315 900, 310 415, 82 350, 441 780, 366 230, 356 265, 352 944, 153 679, 879 588, 10 114,
246 912, 204 315, 90 240, 663 060, 558, 66 132, 443 700, 246 759, 163 928, 161 910,
140 608, 174 966, 26 622, 189 663, 15 072, 64 460, 22 950, 145 921, 90 168, 459 277,
351 372, 22 605, 322 328, 325 926, 224 406, 86 994, 1282, 313 760, 558 279, 273 132,
88 995, 280 014, 214 061, 22 473, 244 278, 59 280, 161 850, 35 154, 689 724, 274 628,
658 176, 94 248, 42 529, 244 188, 236 800, 131 588, 186 214, 424 200, 11 676, 206 468,
85 436, 458 658, 39 693, 818 380, 591 364, 29 202, 271 566, 32 760, 155 574, 24 288,
318 756, 12 276, 545 616, 95 125, 920 178, 604 800, 508 980, 558 297, 315 840, 859 142}

In[31]:=
```
inputEngine // GearRatios // Total
```

Out[31]=

87 605 697