The testing of the 4 methods' effectiveness, on the 100th knot in Dunfield's list of knots.

- First, the knot that we are experimenting on:

```
In[∘]:= << KnotTheory`

Loading KnotTheory` version of February 2, 2020, 10:53:45.2097.
Read more at http://katlas.org/wiki/KnotTheory.
```

```
In[∘]:= Ks = ReadList["http://drorbn.net/AcademicPensieve/People/Dunfield/nmd_random_knots"]
```

```
Out[∘]= {PD[X[3, 1, 4, 0], X[5, 3, 0, 2], X[1, 5, 2, 4]], … 996 …,
         PD[X[555, 537, 556, 536], X[1662, 1693, 1663, 1694], X[422, 426, 423, 425],
          X[1962, 1888, 1963, 1887], X[760, 338, 761, 337], X[331, 790, 332, 791],
          X[783, 775, 784, 774], X[723, 709, 724, 708], X[1728, 1570, 1729, 1569],
          … 983 …, X[1453, 1702, 1454, 1703], X[956, 959, 957, 960],
          X[1429, 478, 1430, 479], X[1257, 1255, 1258, 1254], X[814, 1294, 815, 1293],
          X[1699, 1484, 1700, 1485], X[1156, 1133, 1157, 1134], X[1444, 486, 1445, 485]]}

        large output | show less | show more | show all | set size limit...
```

- the below testing knot is bestGreedy[Ks[[98]], {factor -> JonesResult, tries -> 1000}], performing the algorithm many times might generate different results:

```
In[•]:= testingKnot = {X[111, 106, 112, 107], X[105, 112, 106, 113], X[102, 114, 103, 113],
           X[114, 102, 115, 101], X[115, 111, 116, 110], X[109, 117, 110, 116], X[59, 109, 60, 108],
           X[107, 61, 108, 60], X[64, 103, 65, 104], X[104, 63, 105, 64], X[58, 55, 59, 56],
           X[62, 57, 63, 58], X[56, 61, 57, 62], X[182, 66, 183, 65], X[54, 184, 55, 183],
           X[66, 182, 67, 181], X[67, 101, 68, 100], X[117, 69, 118, 68], X[99, 181, 100, 180],
           X[126, 70, 127, 69], X[127, 119, 128, 118], X[70, 120, 71, 119], X[71, 129, 72, 128],
           X[120, 126, 121, 125], X[184, 122, 185, 121], X[46, 185, 47, 186], X[124, 45, 125, 46],
           X[186, 47, 187, 48], X[122, 188, 123, 187], X[48, 123, 49, 124], X[188, 50, 189, 49],
           X[53, 50, 54, 51], X[51, 99, 52, 98], X[97, 53, 98, 52], X[189, 154, 190, 155],
           X[44, 156, 45, 155], X[156, 130, 157, 129], X[72, 157, 73, 158], X[96, 154, 97, 153],
           X[158, 179, 159, 180], X[133, 130, 134, 131], X[178, 161, 179, 162], X[73, 163, 74, 162],
           X[163, 133, 164, 132], X[131, 165, 132, 164], X[169, 75, 170, 74], X[75, 171, 76, 170],
           X[177, 77, 178, 76], X[171, 169, 172, 168], X[167, 173, 168, 172], X[173, 167, 174, 166],
           X[165, 175, 166, 174], X[79, 176, 80, 177], X[175, 78, 176, 79], X[77, 80, 78, 81],
           X[136, 81, 137, 82], X[82, 137, 83, 138], X[134, 84, 135, 83], X[138, 135, 139, 136],
           X[0, 160, 1, 159], X[160, 0, 161, 199], X[198, 140, 199, 139], X[84, 197, 85, 198],
           X[43, 197, 44, 196], X[190, 195, 191, 196], X[191, 42, 192, 43], X[194, 42, 195, 41],
           X[140, 2, 141, 1], X[85, 3, 86, 2], X[3, 192, 4, 193], X[193, 4, 194, 5],
           X[141, 86, 142, 87], X[152, 88, 153, 87], X[88, 95, 89, 96], X[94, 89, 95, 90],
           X[90, 93, 91, 94], X[92, 20, 93, 19], X[20, 92, 21, 91], X[40, 11, 41, 12],
           X[5, 149, 6, 148], X[149, 11, 150, 10], X[9, 6, 10, 7], X[147, 8, 148, 9], X[7, 146, 8, 147],
           X[142, 146, 143, 145], X[150, 144, 151, 143], X[144, 152, 145, 151], X[18, 23, 19, 24],
           X[24, 40, 25, 39], X[12, 26, 13, 25], X[13, 38, 14, 39], X[28, 22, 29, 21],
           X[22, 28, 23, 27], X[26, 29, 27, 30], X[17, 30, 18, 31], X[31, 16, 32, 17],
           X[37, 33, 38, 32], X[33, 37, 34, 36], X[35, 14, 36, 15], X[15, 34, 16, 35]};
```

- THE ESTIMATED RUN TIMES

```
In[•]:= unselTable = Flatten[Table[bestGreedySamples[PD @@ testingKnot,
             {factor → JonesResult, neededSamples → 1, tries → 1000}], {i, 50}], 1];

In[•]:= unselTableJones = JonesResult[#] & /@ unselTable

Out[•]= {2 263 188, 2 086 110, 2 079 104, 2 223 436, 1 897 210, 2 166 826, 2 268 688, 1 461 010, 1 832 278,
         1 861 372, 1 536 208, 2 255 968, 1 649 546, 2 092 444, 1 822 444, 1 720 520, 1 850 072, 2 136 150,
         1 471 898, 2 166 588, 1 657 782, 1 487 408, 1 823 802, 2 479 510, 1 739 164, 2 106 742,
         1 551 874, 1 772 404, 1 923 472, 1 833 218, 2 721 004, 1 665 432, 2 012 496, 1 462 586,
         2 608 880, 1 680 322, 1 864 232, 2 312 424, 1 552 142, 2 184 904, 1 533 392, 1 395 480,
         1 996 314, 1 845 756, 1 410 132, 1 960 404, 1 783 390, 1 526 550, 1 709 310, 1 513 004}

In[•]:= Mean[unselTableJones] * RunTimeConstant

Out[•]= 24.4282
```

- For the following algorithms, essentially we will test how much time it will take to perform local minimization, and from there we will be able to estimate the total run time of the local minimization algorithm we employ plus the time needed to calculate the Kauffman Bracket after. Note to estimate the total run time of the algorithm and the Kauffman Bracket calculation we will take the average Kauffman Bracket calculation time add it to the total time it takes to conduct the local minimization algorithms on all 50 knots, divided by 50.

■ LM

```
In[∘]:= Timing[localOptimization[PD @@ #, {startPt → convertToList,
              invariantCriteria → JonesOptimize, optSize → 6}] &/@ unselTable]⟦1⟧

Out[∘]= 168.844

In[∘]:= LOunselTable = localOptimization[PD @@ #, {startPt → convertToList,
              invariantCriteria → JonesOptimize, optSize → 6}] &/@ unselTable;

In[∘]:= LOJones = JonesResult[#] &/@ LOunselTable;

In[∘]:= Mean[LOJones] * RunTimeConstant

Out[∘]= 19.6881

In[∘]:= 19.688 + 168.844 / 50

Out[∘]= 23.0649

In[∘]:= Timing[localOptimization[PD @@ #, {startPt → convertToList,
              invariantCriteria → JonesOptimize, optSize → 5}] &/@ unselTable]⟦1⟧

Out[∘]= 24.1875

In[∘]:= LOunselTableSmall = localOptimization[PD @@ #, {startPt → convertToList,
              invariantCriteria → JonesOptimize, optSize → 5}] &/@ unselTable;

In[∘]:= LOJonesSmall = JonesResult[#] &/@ LOunselTableSmall;

In[∘]:= Mean[LOJonesSmall] * RunTimeConstant

Out[∘]= 20.6822

In[∘]:= 20.6822 + 24.1875 / 50

Out[∘]= 21.166
```

■ PLM

```
In[∘]:= Timing[partialLO[PD @@ #,
             {startPt → convertToList, invariantCriteria → PartialJonesOptimize,
              optSize → 6, partialPortion → 0.1, rounds → 10}] &/@ unselTable]⟦1⟧

Out[∘]= 191.703

In[∘]:= PLOunselTable = partialLO[PD @@ #,
             {startPt → convertToList, invariantCriteria → PartialJonesOptimize,
              optSize → 6, partialPortion → 0.1, rounds → 10}] &/@ unselTable;

In[∘]:= PLOJones = JonesResult[#] &/@ PLOunselTable;

In[∘]:= Mean[PLOJones] * RunTimeConstant

Out[∘]= 19.142

In[∘]:= 19.14 + 191.703125 / 50

Out[∘]= 22.9741
```

```
In[•]:= Timing[partialLO[PD @@ #,
              {startPt → convertToList, invariantCriteria → PartialJonesOptimize,
               optSize → 5, partialPortion → 0.1, rounds → 10}] & /@ unselTable][[1]]

Out[•]= 32.3594

In[•]:= PLOunselTableSmall = partialLO[PD @@ #,
              {startPt → convertToList, invariantCriteria → PartialJonesOptimize,
               optSize → 5, partialPortion → 0.1, rounds → 10}] & /@ unselTable;

In[•]:= PLOJonesSmall = JonesResult[#] & /@ PLOunselTableSmall;

In[•]:= Mean[PLOJonesSmall] * RunTimeConstant

Out[•]= 22.5528

In[•]:= 22.553 + 32.359375 / 50

Out[•]= 23.2002

In[•]:= Timing[partialLO[PD @@ #,
              {startPt → convertToList, invariantCriteria → PartialJonesOptimize,
               optSize → 5, partialPortion → 0.2, rounds → 5}] & /@ unselTable][[1]]

Out[•]= 28.2344

In[•]:= PLOunselTableSmallMod = partialLO[PD @@ #,
              {startPt → convertToList, invariantCriteria → PartialJonesOptimize,
               optSize → 5, partialPortion → 0.2, rounds → 5}] & /@ unselTable;

In[•]:= PLOJonesSmallMod = JonesResult[#] & /@ PLOunselTableSmallMod;

In[•]:= Mean[PLOJonesSmallMod] * RunTimeConstant

Out[•]= 19.4038

In[•]:= 19.40 + 28.234375` / 50

Out[•]= 19.9647
```

- RLM

```
In[•]:= Timing[reverseLocalOptimization[PD @@ #, {startPt → convertToList,
               invariantCriteria → JonesOptimize, optSize → 6}] & /@ unselTable][[1]]

Out[•]= 176.234

In[•]:= RLOunselTable = reverseLocalOptimization[PD @@ #, {startPt → convertToList,
               invariantCriteria → JonesOptimize, optSize → 6}] & /@ unselTable;

In[•]:= RLOJones = JonesResult[#] & /@ RLOunselTable;

In[•]:= Mean[RLOJones] * RunTimeConstant

Out[•]= 18.9208
```

```
In[∘]:= 18.9208 + 176.234 / 50

Out[∘]= 22.4455

In[∘]:= Timing[reverseLocalOptimization[PD @@ #, {startPt → convertToList,
              invariantCriteria → JonesOptimize, optSize → 5}] & /@ unselTable][[1]]

Out[∘]= 33.4688

In[∘]:= RLOunselTableSmall = reverseLocalOptimization[PD @@ #, {startPt → convertToList,
              invariantCriteria → JonesOptimize, optSize → 5}] & /@ unselTable;

In[∘]:= RLOJonesSmall = JonesResult[#] & /@ RLOunselTableSmall;

In[∘]:= Mean[RLOJonesSmall] * RunTimeConstant

Out[∘]= 20.2043

In[∘]:= 20.2043 + 33.4688 / 50

Out[∘]= 20.8737
```

- RPLM

```
In[∘]:= Protect[startPt, invariantCriteria, optSize, partialPortion, rounds];

In[∘]:= Options[backwardsPLO] :=
          {startPt → convertToList, invariantCriteria → PartialJonesOptimize,
           optSize → 3, partialPortion → 0.1, rounds → 3};

In[∘]:= backwardsPLO[pd_PD, opts : OptionsPattern[backwardsPLO]] :=
         Module[{s, k, sBest, optimizeList, optimizeResult, widths, width, round},
          s = OptionValue[startPt][pd];
          k = Length[s] - OptionValue[optSize] + 1;
          round = 1;
          sBest = s;
          While[round ≤ OptionValue[rounds],
           widths = FoldList[Complement[#1 ⋃ #2, #1 ⋂ #2] &, {}, List @@@ sBest];
           While[k ≥ 1,
            optimizeList = sBest[[k ;; k + OptionValue[optSize] - 1]];
            width =  widths[[k]];
            optimizeResult = OptionValue[invariantCriteria][
              width, optimizeList, k, OptionValue[partialPortion]];
            sBest = Join[If[k == 1, {}, sBest[[1 ;; k - 1]]], optimizeResult,
              If[k == Length[s] - OptionValue[optSize] + 1,
               {}, sBest[[k + OptionValue[optSize] ;; -1]]]];
            k = k - 1;
           ];
           round = round + 1;
           k  = Length[s] - OptionValue[optSize] + 1;
          ];
          sBest]
```

```
In[∘]:= Timing[backwardsPLO[PD @@ #,
              {startPt → convertToList, invariantCriteria → PartialJonesOptimize,
               optSize → 6, partialPortion → 0.1, rounds → 10}] & /@ unselTable]⟦1⟧

Out[∘]= 181.641

In[∘]:= BPLOunselTable =  backwardsPLO[PD @@ #,
              {startPt → convertToList, invariantCriteria → PartialJonesOptimize,
               optSize → 6, partialPortion → 0.1, rounds → 10}] & /@ unselTable;

In[∘]:= BPLOJones = JonesResult[#] & /@ BPLOunselTable;

In[∘]:= Mean[BPLOJones] * RunTimeConstant

Out[∘]= 17.2924

In[∘]:= 17.2924 + 181.641 / 50

Out[∘]= 20.9252

In[∘]:= Timing[backwardsPLO[PD @@ #,
              {startPt → convertToList, invariantCriteria → PartialJonesOptimize,
               optSize → 5, partialPortion → 0.1, rounds → 10}] & /@ unselTable]⟦1⟧

Out[∘]= 31.3125

In[∘]:= BPLOunselTableSmall =  backwardsPLO[PD @@ #,
              {startPt → convertToList, invariantCriteria → PartialJonesOptimize,
               optSize → 5, partialPortion → 0.1, rounds → 10}] & /@ unselTable;

In[∘]:= BPLOJonesSmall = JonesResult[#] & /@ BPLOunselTableSmall;

In[∘]:= Mean[BPLOJonesSmall] * RunTimeConstant

Out[∘]= 18.6993

In[∘]:= 18.6993 + 31.3125 / 50

Out[∘]= 19.3256

In[∘]:= Timing[backwardsPLO[PD @@ #,
              {startPt → convertToList, invariantCriteria → PartialJonesOptimize,
               optSize → 5, partialPortion → 0.2, rounds → 5}] & /@ unselTable]⟦1⟧

Out[∘]= 27.3125

In[∘]:= BPLOunselTableSmallMod =  backwardsPLO[PD @@ #,
              {startPt → convertToList, invariantCriteria → PartialJonesOptimize,
               optSize → 5, partialPortion → 0.2, rounds → 5}] & /@ unselTable;

In[∘]:= BPLOJonesSmallMod = JonesResult[#] & /@ BPLOunselTableSmallMod;

In[∘]:= Mean[BPLOJonesSmallMod] * RunTimeConstant

Out[∘]= 18.8515
```

```
In[∘]:= 18.8515 + 27.3125 / 50

Out[∘]= 19.3978
```

- From there we will be able to actually calculate the time it takes to calculate the Kauffman Bracket, so it is a repeat of the process above.

```
In[∘]:= unselTableFKB = (# /. Global`X → X) & /@ unselTable;

In[∘]:= Timing[FKB[#]]⟦1⟧ & /@ unselTableFKB

Out[∘]= {30.9219, 40.6406, 40.5781, 39.7656, 35., 30.2344, 36.4375, 32.5938, 35.5156, 24.375,
         25.2344, 43.8125, 33.5781, 46.1406, 35.5313, 32.7031, 39.3594, 32.7344, 22.5, 26.375,
         24.5938, 33.8438, 37.1094, 34.7969, 34.8281, 24.1563, 32.7031, 35.5, 42.75, 23.2344,
         51.4531, 30.6094, 38.8125, 20.6875, 31.4844, 31.5625, 40.5313, 35.5938, 20.9688, 31.6719,
         20.0625, 26.7344, 42.2031, 37.5938, 16.9219, 29.9063, 23.4375, 33.8906, 40.9219, 18.2344}

In[∘]:= Mean[{30.921875`, 40.640625`, 40.578125`, 39.765625`, 35.`, 30.234375`, 36.4375`,
          32.59375`, 35.515625`, 24.375`, 25.234375`, 43.8125`, 33.578125`, 46.140625`,
          35.53125`, 32.703125`, 39.359375`, 32.734375`, 22.5`, 26.375`, 24.59375`, 33.84375`,
          37.109375`, 34.796875`, 34.828125`, 24.15625`, 32.703125`, 35.5`, 42.75`,
          23.234375`, 51.453125`, 30.609375`, 38.8125`, 20.6875`, 31.484375`, 31.5625`,
          40.53125`, 35.59375`, 20.96875`, 31.671875`, 20.0625`, 26.734375`, 42.203125`,
          37.59375`, 16.921875`, 29.90625`, 23.4375`, 33.890625`, 40.921875`, 18.234375`}]

Out[∘]= 32.6166

In[∘]:= FKBTimeLO = Timing[FKB[#]]⟦1⟧ & /@ ((# /. Global`X → X) & /@ LOunselTable);

In[∘]:= Mean[FKBTimeLO]

Out[∘]= 26.5456

In[∘]:= 191.703125 / 50 + 26.545625`

Out[∘]= 30.3797

In[∘]:= FKBTimeLOSmall = Timing[FKB[#]]⟦1⟧ & /@ ((# /. Global`X → X) & /@ LOunselTableSmall);

In[∘]:= Mean[FKBTimeLOSmall]

Out[∘]= 27.6988

In[∘]:= 27.69875 + 24.1875 / 50

Out[∘]= 28.1825

In[∘]:= FKBTimePLO = Timing[FKB[#]]⟦1⟧ & /@ ((# /. Global`X → X) & /@ PLOunselTable);

In[∘]:= Mean[FKBTimePLO]

Out[∘]= 23.7722

In[∘]:= 23.7721875 + 191.703125 / 50

Out[∘]= 27.6063

In[∘]:= FKBTimePLOSmall = Timing[FKB[#]]⟦1⟧ & /@ ((# /. Global`X → X) & /@ PLOunselTableSmall);
```

```
In[∘]:= Mean[FKBTimePLOSmall]

Out[∘]= 27.2038

In[∘]:= 27.20375` + 32.359375 / 50

Out[∘]= 27.8509

In[∘]:= FKBTimePLOSmallMod =
          Timing[FKB[#]]⟦1⟧ & /@ ((# /. Global`X → X) & /@ PLOunselTableSmallMod);

In[∘]:= Mean[FKBTimePLOSmallMod]

Out[∘]= 24.8422

In[∘]:= 24.8421875` + 28.234375` / 50

Out[∘]= 25.4069

In[∘]:= FKBTimeRLO = Timing[FKB[#]]⟦1⟧ & /@ ((# /. Global`X → X) & /@ RLOunselTable);

In[∘]:= Mean[FKBTimeRLO]

Out[∘]= 24.4744

In[∘]:= 24.474375 + 176.234 / 50

Out[∘]= 27.9991

In[∘]:= FKBTimeRLOSmall = Timing[FKB[#]]⟦1⟧ & /@ ((# /. Global`X → X) & /@ RLOunselTableSmall);

In[∘]:= Mean[FKBTimeRLOSmall]

Out[∘]= 26.8984

In[∘]:= 26.8984375 + 33.4688 / 50

Out[∘]= 27.5678

In[∘]:= FKBTimeBPLO = Timing[FKB[#]]⟦1⟧ & /@ ((# /. Global`X → X) & /@ BPLOunselTable);

In[∘]:= Mean[FKBTimeBPLO]

Out[∘]= 22.8081

In[∘]:= 22.808125 + 181.641 / 50

Out[∘]= 26.4409

In[∘]:= FKBTimeBPLOSmall =
          Timing[FKB[#]]⟦1⟧ & /@ ((# /. Global`X → X) & /@ BPLOunselTableSmall);

In[∘]:= Mean[FKBTimeBPLOSmall]

Out[∘]= 24.5734

In[∘]:= 24.5734375 + 31.3125 / 50

Out[∘]= 25.1997
```

```
In[◦]:= FKBTimeBPLOSmallMod =
          Timing[FKB[#]][[1]] & /@ ((# /. Global`X → X) & /@ BPLOunselTableSmallMod);

In[◦]:= Mean[FKBTimeBPLOSmallMod]

Out[◦]= 24.9603

In[◦]:= 24.9603125` + 27.3125 / 50

Out[◦]= 25.5066
```