

Advent of Code 2021 - Day 14: Extended Polymerization

Parsing

```
In[44]:= input = "NNCB

CH→B
HH→N
CB→H
NH→C
HB→C
HC→B
HN→C
NN→C
BH→H
NC→B
NB→B
BN→B
BB→N
BC→B
CC→N
CN→C";
input = ReadString[NotebookDirectory[] ~~ "input14"]

In[46]:= lines = StringSplit[input, "\n"]

In[47]:= polymertemplate = Characters@lines[[1]]

Out[47]= {P, P, F, C, H, P, F, N, C, K, O, K, O, S, B, V, C, F, P, P}

In[48]:= stringrules = lines[[3 ;;]]

In[49]:= rules =
  Map[(Characters@StringTake[#, 2] → {{StringTake[#, {1}], StringTake[#, {-1}]},
    {StringTake[#, {-1}], StringTake[#, {2}]}}) &, stringrules] // Sort
```

Create Matrix of Pair Insertion


```
In[50]:= intmapping = MapIndexed[#1 → First@#2 &, rules[[All, 1]]]
```

```
In[ ]:= {"B", "B"} → {{ "B", "N"}, {"N", "B"}} /. intmapping
```

```
Out[ ]:= 1 → {4, 13}
```

```
In[ ]:= transformations =  
  Map[{{#[[2, 1]], #[[1]]}, {#[[2, 2]], #[[1]]}} &, rules] /. intmapping //  
  Flatten[#, 1] &
```

```
In[ ]:= matrix = SparseArray[  
  transformations → Table[1, Length@transformations], {Length@rules, Length@rules}]
```

```
Out[ ]:= SparseArray[  
   Specified elements: 200  
  Dimensions: {100, 100}
```

```
In[ ]:= MatrixForm[matrix]
```

Initial Polymer

```
In[ ]:= Rule @@ # & /@ (Tally@Partition[Characters[lines[[1]]], 2, 1] /. intmapping)
```

```
Out[ ]:= {16 → 1, 14 → 1, 5 → 1}
```

```
In[ ]:= init = Table[0, Length@rules] //  
  ReplacePart[Rule @@ # & /@ (Tally@Partition[polymertemplate, 2, 1] /. intmapping)]
```

Multiplication

```
In[ ]:= resultingpairs = MatrixPower[matrix, 10, init] * rules[[All, 1]]  
doubletotal =  
  Total@Flatten@resultingpairs + First@polymertemplate + Last@polymertemplate
```

```
Out[ ]:= {{0, 0}, {205 B, 205 C}, {207 B, 207 F}, {89 B, 89 H}, {406 B, 406 K}, {872 B, 872 N},  
  {0, 0}, {0, 0}, {143 B, 143 S}, {0, 0}, {548 C, 548 B}, {0, 0}, {134 C, 134 F}, {0, 0},  
  {50 C, 50 K}, {260 C, 260 N}, {65 C, 65 O}, {26 C, 26 P}, {275 C, 275 S}, {48 C, 48 V},  
  {0, 0}, {260 F, 260 C}, {274 F, 274 F}, {0, 0}, {573 F, 573 K}, {537 F, 537 N},  
  {83 F, 83 O}, {0, 0}, {259 F, 259 S}, {176 F, 176 V}, {195 H, 195 B}, {77 H, 77 C},  
  {104 H, 104 F}, {58 H, 58 H}, {16 H, 16 K}, {29 H, 29 N}, {365 H, 365 O}, {317 H, 317 P},  
  {216 H, 216 S}, {0, 0}, {91 K, 91 B}, {187 K, 187 C}, {209 K, 209 F}, {48 K, 48 H}, {0, 0},  
  {199 K, 199 N}, {433 K, 433 O}, {82 K, 82 P}, {0, 0}, {535 K, 535 V}, {303 N, 303 B},  
  {391 N, 391 C}, {436 N, 436 F}, {29 N, 29 H}, {633 N, 633 K}, {423 N, 423 N}, {302 N, 302 O},  
  {105 N, 105 P}, {0, 0}, {416 N, 416 V}, {217 O, 217 B}, {0, 0}, {280 O, 280 F},  
  {682 O, 682 H}, {0, K}, {432 O, 432 N}, {568 O, 568 O}, {401 O, 401 P}, {257 O, 257 S},  
  {239 O, 239 V}, {293 P, 293 B}, {120 P, 120 C}, {83 P, 83 F}, {95 P, 95 H}, {34 P, 34 K},  
  {22 P, 22 N}, {166 P, 166 O}, {0, 0}, {109 P, 109 S}, {127 P, 127 V}, {275 S, 275 B},  
  {166 S, 166 C}, {0, 0}, {366 S, 366 H}, {71 S, 71 K}, {56 S, 56 N}, {183 S, 183 O},  
  {92 S, 92 P}, {362 S, 362 S}, {50 S, 50 V}, {0, 0}, {0, 0}, {435 V, 435 F}, {10 V, 10 H},  
  {0, 0}, {208 V, 208 N}, {912 V, 912 O}, {26 V, 26 P}, {0, 0}, {429 V, 429 V}}
```

```
Out[ ]:= 3844 B + 2812 C + 4324 F + 2754 H + 3568 K + 6076 N + 6154 O + 2100 P + 3242 S + 4040 V
```

```
In[6]:= #[[1]] / 2 & /@List@@doubletotal // MinMax // Differences // First
```

```
Out[6]= 2027
```

```
In[7]:= polymerization[input_, step_] :=
Block[{lines = StringSplit[input, "\n"], polymertemplate, stringrules, rules,
  intmapping, transformations, matrix, init, resultingpairs, doubletotal},
  polymertemplate = Characters@lines[[1]];
  stringrules = lines[[3 ;;]];
  rules =
    Map[(Characters@StringTake[#, 2] → {{StringTake[#, {1}], StringTake[#, {-1}]},
      {StringTake[#, {-1}], StringTake[#, {2}]}}] &, stringrules] // Sort;
  intmapping = MapIndexed[#1 → First@#2 &, rules][[All, 1]];
  transformations =
    (Map[{{#[[2, 1]], #[[1]]}, {#[[2, 2]], #[[1]]}} &, rules] /. intmapping //
      Flatten[#, 1] &);
  matrix = SparseArray[transformations → Table[1, Length@transformations],
    {Length@rules, Length@rules}];
  init = Table[0, Length@rules] // ReplacePart[
    Rule@@# & /@ (Tally@Partition[polymertemplate, 2, 1] /. intmapping)];
  resultingpairs = MatrixPower[matrix, step, init] * rules[[All, 1]];
  doubletotal =
    Total@Flatten@resultingpairs + First@polymertemplate + Last@polymertemplate;
  #[[1]] / 2 & /@List@@doubletotal // MinMax // Differences // First
]
```

```
In[47]:= lines = StringSplit[ReadString[NotebookDirectory[] ~~ "input14"], "\n"];
First@Differences@MinMax@(#[[1]] / 2 & /@
  List@@(Total@Flatten@(MatrixPower[SparseArray[# → Table[1, Length@#] &@
    (Map[{{#[[2, 1]], #[[1]]}, {#[[2, 2]], #[[1]]}} &, #] /.
      MapIndexed[#1 → First@#2 &, #[[All, 1]]] // Flatten[#, 1] &),
    {Length@#, Length@#}], 40, Table[0, Length@#] // ReplacePart[
      Rule@@# & /@ (Tally@Partition[Characters@lines[[1]], 2, 1] /.
        MapIndexed[#1 → First@#2 &, #[[All, 1]]]) * #[[All, 1]] +
        First@Characters@lines[[1]] + Last@Characters@lines[[1]]) &@
    Sort[Map[(Characters@StringTake[#, 2] → {{StringTake[#, {1}],
      StringTake[#, {-1}]},
      {StringTake[#, {-1}], StringTake[#, {2}]}}] &, lines[[3 ;;]])]
```

```
Out[48]= 2 265 039 461 737
```

```
In[49]:= function = "lines=StringSplit[ReadString[NotebookDirectory[] ~~ \"input14\"], \"\\n\"];
  First@Differences@MinMax@(#[[1]] / 2 & /@List@@(Total@Flatten@(MatrixPower[
  SparseArray[#→Table[1,Length@#]&@(Map[{{#[[2,1]],#[[1]]},{#[[2,2]],#[[1]]}}&,#
  ] /.MapIndexed[#1→First@#2&,[All,1]]] //Flatten[#,1]&),{Length@#,Length@#}],40,
  Table[0,Length@#] //ReplacePart[Rule@@#&/@ (Tally@Partition[Characters@lines[[1]],
  2,1] /.MapIndexed[#1→First@#2&,[All,1]]]) *#[[All,1]]+First@Characters@lines
  [[1]]+Last@Characters@lines[[1]])&@Sort[Map[(Characters@StringTake[#,2]→{{
  StringTake[#,1],StringTake[#{-1}],{StringTake[#{-1}],StringTake[#{2}]]}}&
  ,lines[[3;;]])]"
```

In[65]:= **Compress@function**

```
Out[65]= 1:eJyNkV9rwjAUxWXfZBZK07PVRvcyGUQ2hIE0sb7FPES9ajAkJc2YUvQL7EsvNW70/YE93tPTc3/35
HKqR9nbRa0mhYLiPrNGqGWS2HpCPjcz/RZW5hqvX4UBmZWmy1l+31dqPzFpu06Q/VJM8V11pnc9VrdVk+
YwpJHsViAATWDggyEGvANiQJKU8YSHCakL5yHRGNtuSQ9ya0FRaIBdws3Q/0
Khmy5NwV0jeFbGrjkw4zHfCqBpqgPam1XJGBh9Ut0y9I1Y+SykV+xQ17BJ2UXooA1N87+p0awgTkNUP/
qeQMcvtau1FU0S5IjFA3cHMAo/
Fj6uX3HULuJPFpPzxJQkI8glN8GQG9fiiwRCAndxNOZSbkm1Ciu0og8rbvjMginIof0DJ6r0+
D9mzNjVaYwb3vRrcqPP//oUxyHJtK0tyoy+WPz7j/kaXA2YeZCyPJPL1BVxr1w7yT3BT/
GbDzvbLg7RkaTV6biT2DtZGt03
```

In[66]:= **ToExpression@Uncompress@**

```
"1:eJyNkV9rwjAUxWXfZBZK07PVRvcyGUQ2hIE0sb7FPES9ajAkJc2YUvQL7EsvNW70/YE93tPTc3/35
HKqR9nbRa0mhYLiPrNGqGWS2HpCPjcz/RZW5hqvX4UBmZWmy1l+31dqPzFpu06Q/
VJM8V11pnc9VrdVk+YwpJHsViAATWDggyEGvANiQJKU8YSHCakL5yHRGNtuSQ9ya0FRaIBdws3Q/0
Khmy5NwV0jeFbGrjkw4zHfCqBpqgPam1XJGBh9Ut0y9I1Y+SykV+xQ17BJ2UXooA1N87+p0awgTkNUP/
/qeQMcvtau1FU0S5IjFA3cHMAo/
Fj6uX3HULuJPFpPzxJQkI8glN8GQG9fiiwRCAndxNOZSbkm1Ciu0og8rbvjMginIof0DJ6r0+
D9mzNjVaYwb3vRrcqPP//oUxyHJtK0tyoy+WPz7j/kaXA2YeZCyPJPL1BVxr1w7yT3BT/
GbDzvbLg7RkaTV6biT2DtZGt03"
```

Out[66]= 2 265 039 461 737

In[64]:= **ByteCount["ToExpression@Uncompress@"1:**

```
eJyNkV9rwjAUxWXfZBZK07PVRvcyGUQ2hIE0sb7FPES9ajAkJc2YUvQL7EsvNW70/YE93tPTc3/35
HKqR9nbRa0mhYLiPrNGqGWS2HpCPjcz/RZW5hqvX4UBmZWmy1l+31dqPzFpu06Q/VJM8V11pnc9VrdVk+
YwpJHsViAATWDggyEGvANiQJKU8YSHCakL5yHRGNtuSQ9ya0FRaIBdws3Q/0
Khmy5NwV0jeFbGrjkw4zHfCqBpqgPam1XJGBh9Ut0y9I1Y+SykV+xQ17BJ2UXooA1N87+p0awgTkNUP/
qeQMcvtau1FU0S5IjFA3cHMAo/
Fj6uX3HULuJPFpPzxJQkI8glN8GQG9fiiwRCAndxNOZSbkm1Ciu0og8rbvjMginIof0DJ6r0+
D9mzNjVaYwb3vRrcqPP//oUxyHJtK0tyoy+WPz7j/kaXA2YeZCyPJPL1BVxr1w7yT3BT/
GbDzvbLg7RkaTV6biT2DtZGt03\[""]]
```

Out[64]= 576

In[18]:= **RepeatedTiming[polymerization[input, 40], 1]**

Out[18]= {0.0046314, 2 265 039 461 737}

In[]:= **For[step = 32000, step < 100000, step += 1000,
polymerization[input, Echo@step] * 1.0 // AbsoluteTiming // Echo]**

```
» 32000
» {0.663232, 1.878258947114374 × 109633}
» 33000
» {1.58618, 2.012570628357668 × 109934}
» 34000
» {1.63862, 2.156486750855516 × 1010235}
» 35000
» {1.63067, 2.310694114824833 × 1010536}
» 36000
» {1.58703, 2.475928632609461 × 1010837}
» 37000
```

- » {1.62063, 2.652978840619963 $\times 10^{1138}$ }
- » 38 000
- » {1.62801, 2.842689662407335 $\times 10^{11439}$ }
- » 39 000
- » {1.6444, 3.045966440828884 $\times 10^{11740}$ }
- » 40 000
- » {1.60855, 3.263779258548670 $\times 10^{12041}$ }
- » 41 000
- » {1.58829, 3.497167567490913 $\times 10^{12342}$ }
- » 42 000
- » {1.59745, 3.747245149339175 $\times 10^{12643}$ }
- » 43 000
- » {1.61218, 4.015205430753916 $\times 10^{12944}$ }
- » 44 000
- » {1.61584, 4.302327178673865 $\times 10^{13245}$ }
- » 45 000
- » {1.6492, 4.609980602880456 $\times 10^{13546}$ }
- » 46 000
- » {1.63512, 4.939633894948146 $\times 10^{13847}$ }
- » 47 000
- » {1.64544, 5.292860234785965 $\times 10^{14148}$ }
- » 48 000
- » {1.65824, 5.671345298207089 $\times 10^{14449}$ }
- » 49 000
- » {1.65428, 6.076895301354264 $\times 10^{14750}$ }
- » 50 000
- » {1.61782, 6.511445620370879 $\times 10^{15051}$ }
- » 51 000
- » {1.61168, 6.977070027452721 $\times 10^{15352}$ }
- » 52 000
- » {1.60583, 7.475990587356917 $\times 10^{15653}$ }
- » 53 000
- » {1.63843, 8.010588261596455 $\times 10^{15954}$ }
- » 54 000
- » {1.65888, 8.583414270925880 $\times 10^{16255}$ }
- » 55 000
- » {1.63687, 9.19720227034252 $\times 10^{16556}$ }
- » 56 000

- » {1.63233, 9.85488139470509 $\times 10^{16857}$ }
- » 57 000
- » {1.6716, 1.055959023722631 $\times 10^{17159}$ }
- » 58 000
- » {1.63974, 1.131469182754807 $\times 10^{17460}$ }
- » 59 000
- » {1.63466, 1.212378968087787 $\times 10^{17761}$ }
- » 60 000
- » {1.63549, 1.299074499477669 $\times 10^{18062}$ }
- » 61 000
- » {1.63004, 1.391969507566514 $\times 10^{18363}$ }
- » 62 000
- » {1.6311, 1.491507308298350 $\times 10^{18664}$ }
- » 63 000
- » {1.65315, 1.598162918522904 $\times 10^{18965}$ }
- » 64 000
- » {1.68501, 1.712445322883217 $\times 10^{19266}$ }
- » 65 000
- » {1.69849, 1.834899902805234 $\times 10^{19567}$ }
- » 66 000
- » {4.20723, 1.966111039181053 $\times 10^{19868}$ }
- » 67 000
- » {4.40914, 2.106704901166435 $\times 10^{20169}$ }
- » 68 000
- » {4.34997, 2.257352434401330 $\times 10^{20470}$ }
- » 69 000
- » {4.25582, 2.418772562913898 $\times 10^{20771}$ }
- » 70 000
- » {4.17575, 2.591735619988229 $\times 10^{21072}$ }
- » 71 000
- » {4.21479, 2.777067024368625 $\times 10^{21373}$ }
- » 72 000
- » {4.18001, 2.975651219344137 $\times 10^{21674}$ }
- » 73 000
- » {4.15812, 3.188435893511554 $\times 10^{21975}$ }
- » 74 000
- » {4.16627, 3.416436503359267 $\times 10^{22276}$ }
- » 75 000

- » {4.2429, 3.660741119254810 $\times 10^{22577}$ }
- » 76 000
- » {4.20511, 3.922515617962219 $\times 10^{22878}$ }
- » 77 000
- » {4.20315, 4.203009246469077 $\times 10^{23179}$ }
- » 78 000
- » {4.22122, 4.503560583675084 $\times 10^{23480}$ }
- » 79 000
- » {4.24921, 4.825603928392663 $\times 10^{23781}$ }
- » 80 000
- » {4.24338, 5.170676144144602 $\times 10^{24082}$ }
- » 81 000
- » {4.15981, 5.540423993423642 $\times 10^{24383}$ }
- » 82 000
- » {4.15292, 5.936611996414743 $\times 10^{24684}$ }
- » 83 000
- » {4.22418, 6.361130851683647 $\times 10^{24985}$ }
- » 84 000
- » {4.1685, 6.816006459017139 $\times 10^{25286}$ }
- » 85 000
- » {4.21835, 7.303409587474057 $\times 10^{25587}$ }
- » 86 000
- » {4.18733, 7.825666234785158 $\times 10^{25888}$ }
- » 87 000
- » {4.20382, 8.385268727539245 $\times 10^{26189}$ }
- » 88 000
- » {4.2337, 8.984887615128141 $\times 10^{26490}$ }
- » 89 000
- » {4.35198, 9.62738441421110 $\times 10^{26791}$ }
- » 90 000
- » {4.22661, 1.031582526451811 $\times 10^{27093}$ }
- » 91 000
- » {4.2052, 1.105349556116071 $\times 10^{27394}$ }
- » 92 000
- » {4.37573, 1.184391563327890 $\times 10^{27695}$ }
- » 93 000
- » {4.25907, 1.269085754384633 $\times 10^{27996}$ }
- » 94 000

- » {4.24276, 1.359836309080612 $\times 10^{28\,297}$ }
- » 95 000
- » {4.22163, 1.457076309544281 $\times 10^{28\,598}$ }
- » 96 000
- » {4.21601, 1.561269807003899 $\times 10^{28\,899}$ }
- » 97 000
- » {4.32766, 1.672914036344720 $\times 10^{29\,200}$ }
- » 98 000
- » {4.27137, 1.792541789026087 $\times 10^{29\,501}$ }
- » 99 000
- » {4.20844, 1.920723955682523 $\times 10^{29\,802}$ }