

# Advent of Code 2021 - Day 14:

## Extended Polymerization

---

### Parsing

In[20]:=

```
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[
  File["C:\\Users\\Martin\\Documents\\GitHub\\AdventOfCode\\2021\\input14"] ]
```

Out[21]= PPFCHPFNCKOKOSBVCFPF

```
VC -> N
SC -> H
CK -> P
OK -> O
KV -> O
HS -> B
OH -> O
VN -> F
FS -> S
ON -> B
OS -> H
PC -> B
BP -> O
OO -> N
BF -> K
CN -> B
FK -> F
NP -> K
KK -> H
```

CB -> S  
CV -> K  
VS -> F  
SF -> N  
KB -> H  
KN -> F  
CP -> V  
BO -> N  
SS -> O  
HF -> H  
NN -> F  
PP -> O  
VP -> H  
BB -> K  
VB -> N  
OF -> N  
SH -> S  
PO -> F  
OC -> S  
NS -> C  
FH -> N  
FP -> C  
SO -> P  
VK -> C  
HP -> O  
PV -> S  
HN -> K  
NB -> C  
NV -> K  
NK -> B  
FN -> C  
VV -> N  
BN -> N  
BH -> S  
FO -> V  
PK -> N  
PS -> O  
CO -> K  
NO -> K  
SV -> C  
KO -> V  
HC -> B  
BC -> N  
PB -> C  
SK -> S  
FV -> K  
HO -> O  
CF -> O  
HB -> P  
SP -> N  
VH -> P  
NC -> K  
KC -> B  
OV -> P  
BK -> F

```

FB -> F
FF -> V
CS -> F
CC -> H
SB -> C
VO -> V
VF -> O
KP -> N
HV -> H
PF -> H
KH -> P
KS -> S
BS -> H
PH -> S
SN -> K
HK -> P
FC -> N
PN -> S
HH -> N
OB -> P
BV -> S
KF -> N
OP -> H
NF -> V
CH -> K
NH -> P

```

```
In[3]:= lines = StringSplit[input, "\n"]
```

```
Out[3]= {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}
```

```
In[4]:= polymertemplate = Characters@lines[[1]]
```

```
Out[4]= {N, N, C, B}
```

```
In[5]:= stringrules = lines[[3 ;;]]
```

```
Out[5]= {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}
```

```
In[6]:= rules = Map[(Characters@StringTake[#, 2] -> {{StringTake[#, {1}], StringTake[#, {4}]},
               {StringTake[#, {4}], StringTake[#, {2}]}}) &, stringrules] // Sort
```

```
Out[6]= {{B, B} -> {{B, N}, {N, B}}, {B, C} -> {{B, B}, {B, C}},
        {B, H} -> {{B, H}, {H, H}}, {B, N} -> {{B, B}, {B, N}},
        {C, B} -> {{C, H}, {H, B}}, {C, C} -> {{C, N}, {N, C}}, {C, H} -> {{C, B}, {B, H}},
        {C, N} -> {{C, C}, {C, N}}, {H, B} -> {{H, C}, {C, B}}, {H, C} -> {{H, B}, {B, C}},
        {H, H} -> {{H, N}, {N, H}}, {H, N} -> {{H, C}, {C, N}}, {N, B} -> {{N, B}, {B, B}},
        {N, C} -> {{N, B}, {B, C}}, {N, H} -> {{N, C}, {C, H}}, {N, N} -> {{N, C}, {C, N}}}
```

## Create Matrix of Pair Insertion

```
In[7]:= intmapping = MapIndexed[#1 → First@#2 &, rules][[All, 1]]]
```

```
Out[7]= {{B, B} → 1, {B, C} → 2, {B, H} → 3, {B, N} → 4, {C, B} → 5,
         {C, C} → 6, {C, H} → 7, {C, N} → 8, {H, B} → 9, {H, C} → 10, {H, H} → 11,
         {H, N} → 12, {N, B} → 13, {N, C} → 14, {N, H} → 15, {N, N} → 16}
```

```
In[8]:= {"B", "B"} → {{{"B", "N"}, {"N", "B"}}} /. intmapping
```

```
Out[8]= 1 → {4, 13}
```

```
In[8]:= transformations =
  Map[{{#[[2, 1]], #[[1]]}, {#[[2, 2]], #[[1]]}} &, rules] /. intmapping //
  Flatten[#, 1] &
```

```
Out[8]= {{4, 1}, {13, 1}, {1, 2}, {2, 2}, {3, 3}, {11, 3}, {1, 4}, {4, 4},
         {7, 5}, {9, 5}, {8, 6}, {14, 6}, {5, 7}, {3, 7}, {6, 8}, {8, 8}, {10, 9},
         {5, 9}, {9, 10}, {2, 10}, {12, 11}, {15, 11}, {10, 12}, {8, 12}, {13, 13},
         {1, 13}, {13, 14}, {2, 14}, {14, 15}, {7, 15}, {14, 16}, {8, 16}}
```

```
In[9]:= matrix = SparseArray[
  transformations → Table[1, Length@transformations], {Length@rules, Length@rules}]
```

```
Out[9]= SparseArray[ Specified elements: 32  
Dimensions: {16, 16}]
```

```
In[29]:= MatrixForm[matrix]
```

```
Out[29]//MatrixForm=
```

```
( 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 )
( 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 )
( 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 )
( 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 )
( 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 )
( 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 )
( 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 )
( 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 1 )
( 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 )
( 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 )
( 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 )
( 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 )
( 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 )
( 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 )
( 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 )
( 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 )
```

## Initial Polymer

```
In[8]:= Rule@@# & /@ (Tally@Partition[Characters[lines[[1]]], 2, 1] /. intmapping)
```

```
Out[8]= {16 → 1, 14 → 1, 5 → 1}
```

```
In[10]:= init = Table[0, Length@rules] //
  ReplacePart[Rule @@ # & /@ (Tally@Partition[polymertemplate, 2, 1] /. intmapping)]

Out[10]= {0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1}
```

## Multiplication

```
In[11]:= resultingpairs = MatrixPower[matrix, 10, init] * rules[[All, 1]]
doubletotal =
  Total@Flatten@resultingpairs + First@polymertemplate + Last@polymertemplate
```

```
Out[11]= {{812 B, 812 B}, {120 B, 120 C}, {81 B, 81 H}, {735 B, 735 N}, {115 C, 115 B},
  {60 C, 60 C}, {21 C, 21 H}, {102 C, 102 N}, {26 H, 26 B}, {76 H, 76 C},
  {32 H, 32 H}, {27 H, 27 N}, {796 N, 796 B}, {42 N, 42 C}, {27 N, 27 H}, {0, 0}}

Out[12]= 3498 B + 596 C + 322 H + 1730 N
```

```
In[13]:= #[[1]] / 2 & /@List@@doubletotal // MinMax // Differences // First
```

```
Out[13]= 1588
```

```
In[23]:= polymerization[input_, step_] :=
  Module[{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[32]:= polymerization[input, 40] // AbsoluteTiming
```

```
Out[32]= {0.0047535, 2 265 039 461 737}
```

```
In[30]:= For[step = 32000, step < 100000, step += 1000,
  polymerization[input, Echo@step] * 1.0 // AbsoluteTiming // Echo]
```

```
» 32000
```

```
» {0.663232, 1.878258947114374 × 109633}
```

- » 33 000
- »  $\{1.58618, 2.012570628357668 \times 10^{9934}\}$
- » 34 000
- »  $\{1.63862, 2.156486750855516 \times 10^{10235}\}$
- » 35 000
- »  $\{1.63067, 2.310694114824833 \times 10^{10536}\}$
- » 36 000
- »  $\{1.58703, 2.475928632609461 \times 10^{10837}\}$
- » 37 000
- »  $\{1.62063, 2.652978840619963 \times 10^{11138}\}$
- » 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
- »  $\{1.60583, 7.475990587356917 \times 10^{15\,653}\}$
- » 53 000
- »  $\{1.63843, 8.010588261596455 \times 10^{15\,954}\}$
- » 54 000
- »  $\{1.65888, 8.583414270925880 \times 10^{16\,255}\}$
- » 55 000
- »  $\{1.63687, 9.19720227034252 \times 10^{16\,556}\}$
- » 56 000
- »  $\{1.63233, 9.85488139470509 \times 10^{16\,857}\}$
- » 57 000
- »  $\{1.6716, 1.055959023722631 \times 10^{17\,159}\}$
- » 58 000
- »  $\{1.63974, 1.131469182754807 \times 10^{17\,460}\}$
- » 59 000
- »  $\{1.63466, 1.212378968087787 \times 10^{17\,761}\}$
- » 60 000
- »  $\{1.63549, 1.299074499477669 \times 10^{18\,062}\}$
- » 61 000
- »  $\{1.63004, 1.391969507566514 \times 10^{18\,363}\}$
- » 62 000
- »  $\{1.6311, 1.491507308298350 \times 10^{18\,664}\}$
- » 63 000
- »  $\{1.65315, 1.598162918522904 \times 10^{18\,965}\}$
- » 64 000
- »  $\{1.68501, 1.712445322883217 \times 10^{19\,266}\}$
- » 65 000
- »  $\{1.69849, 1.834899902805234 \times 10^{19\,567}\}$
- » 66 000
- »  $\{4.20723, 1.966111039181053 \times 10^{19\,868}\}$
- » 67 000
- »  $\{4.40914, 2.106704901166435 \times 10^{20\,169}\}$
- » 68 000
- »  $\{4.34997, 2.257352434401330 \times 10^{20\,470}\}$
- » 69 000
- »  $\{4.25582, 2.418772562913898 \times 10^{20\,771}\}$
- » 70 000
- »  $\{4.17575, 2.591735619988229 \times 10^{21\,072}\}$

- » 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
- »  $\{4.22661, 1.031582526451811 \times 10^{27\,093}\}$
- » 91 000
- »  $\{4.2052, 1.105349556116071 \times 10^{27\,394}\}$
- » 92 000
- »  $\{4.37573, 1.184391563327890 \times 10^{27\,695}\}$
- » 93 000
- »  $\{4.25907, 1.269085754384633 \times 10^{27\,996}\}$
- » 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
- »  $\{4.20844, 1.920723955682523 \times 10^{29\,802}\}$