Nuclear Structure: Final Exam

Josh Bradt
Spring 2015

Project

- Project A: shell model code
- Goals:

1. Write a shell model code
2. Calculate levels for ^{18}O through ^{28}O
3. Compare to NuShellX and experiment

Outline of code

- Written in Python
- Represents Slater determinants as lists
- M-scheme basis

States & interaction

- Read from file
- USDB interaction
- 1s0d shell states

```
def load_interaction(filename):

    let states be an empty list
    let mels be an empty dictionary

    open file called filename
        for line in file:
            if line starts with '#':
                ignore and continue
            else if line has 6 numbers:
                append list of numbers to states
            else if line has 5 numbers:
                let indices be a tuple of the first 4 numbers
                add entry {indices : last number in line} to mels
    return states, mels
```

Slater determinants

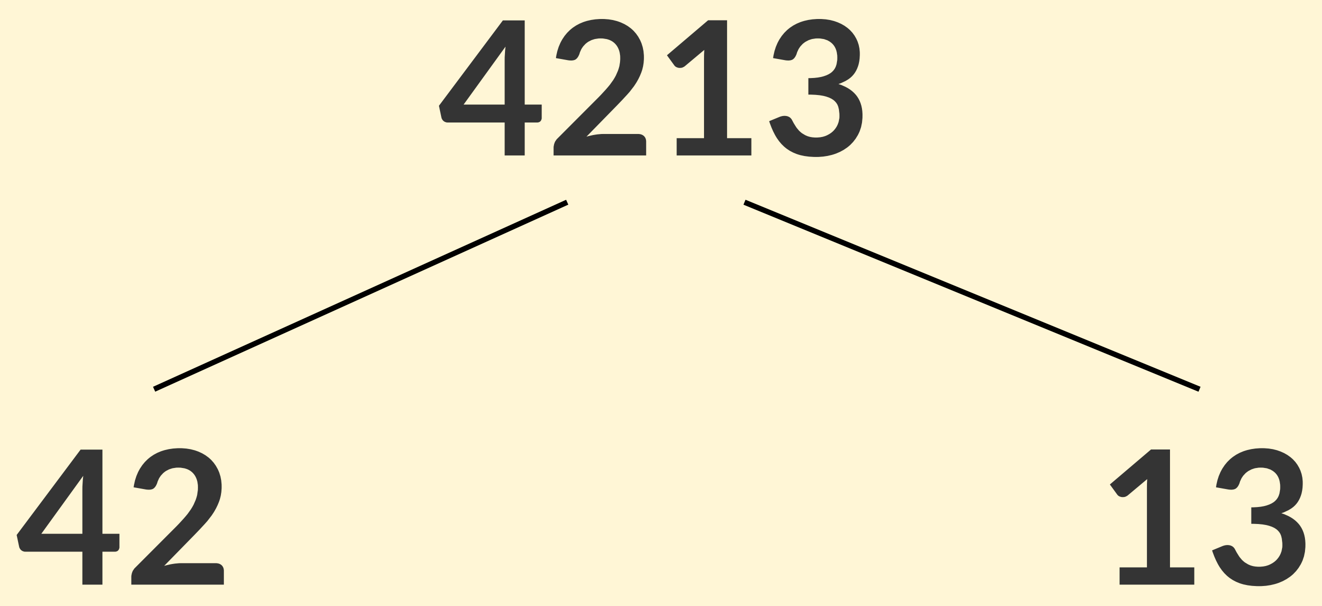
- Find all combinations of indices
    - [0, 1, 2, 3], [0, 1, 2, 4], ...
- Store only those with the chosen total angular momentum projection

```
def slater(n, states, m):

    label the states by index

    let sds be an empty list

    for each unique combination of n states:
        find total M for these states
        if this M == m:
            add indices to sds

    return sds
```

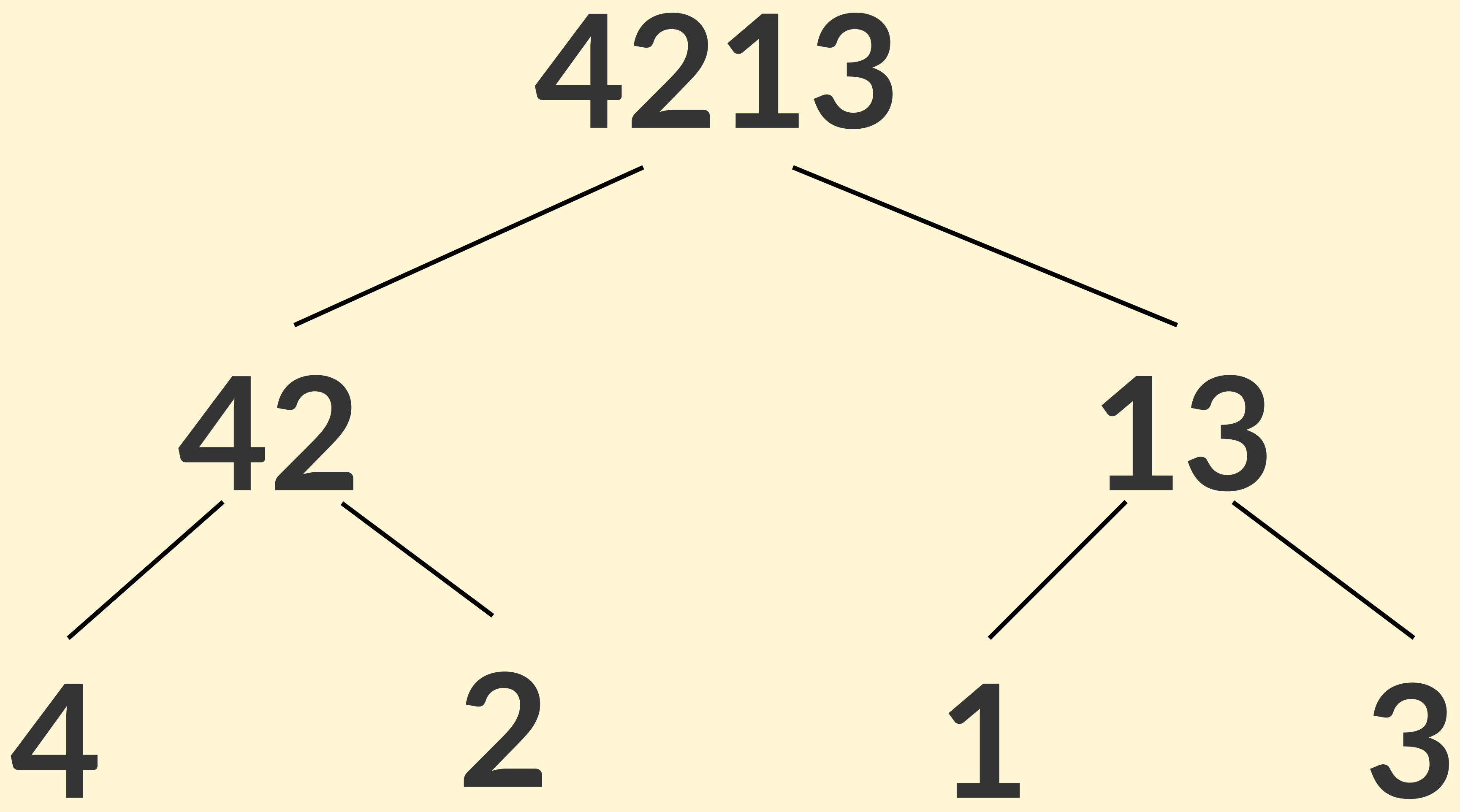
Aside: Finding the Phase

4213

Aside: Finding the Phase



Aside: Finding the Phase



4213

42

13

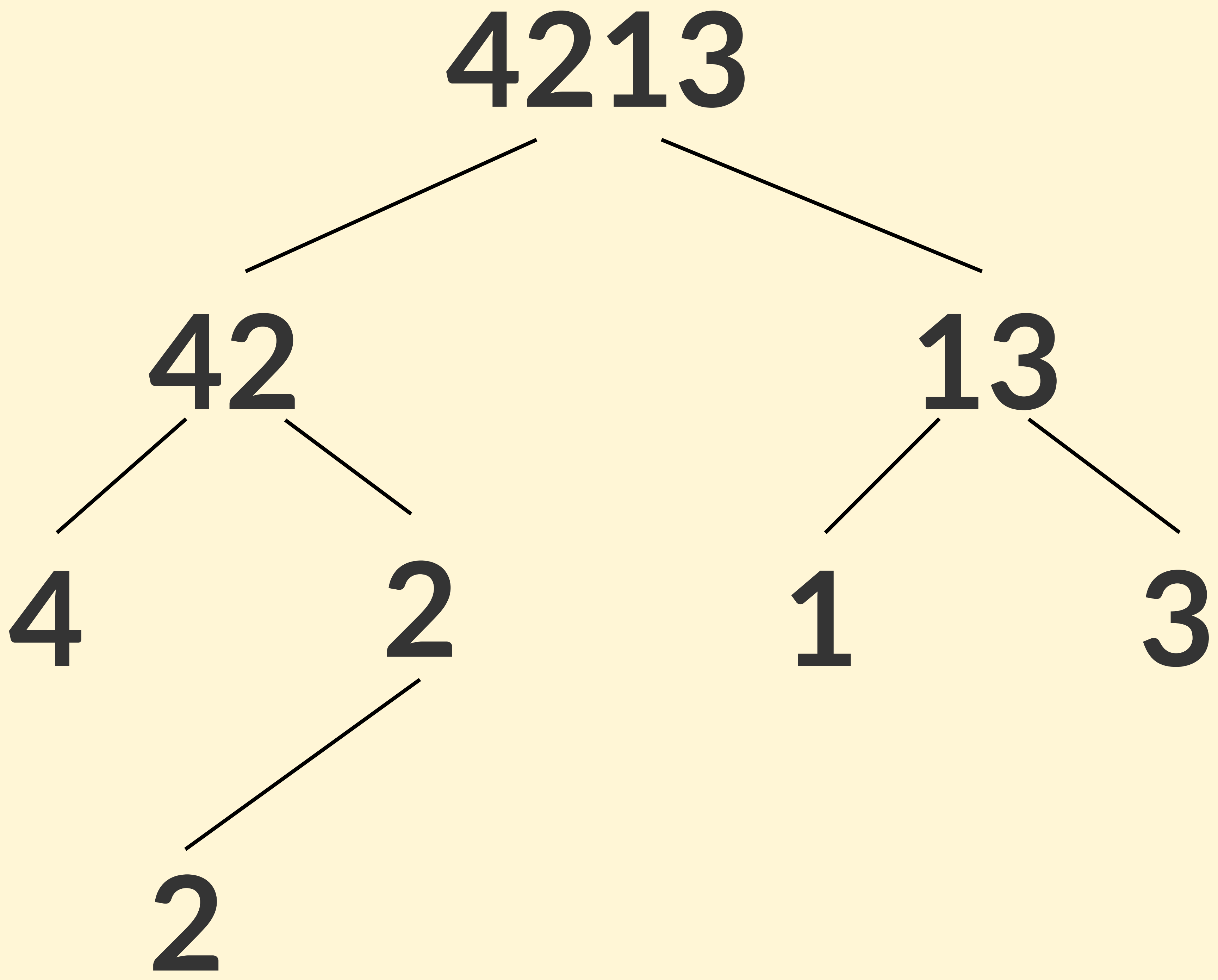
4

2

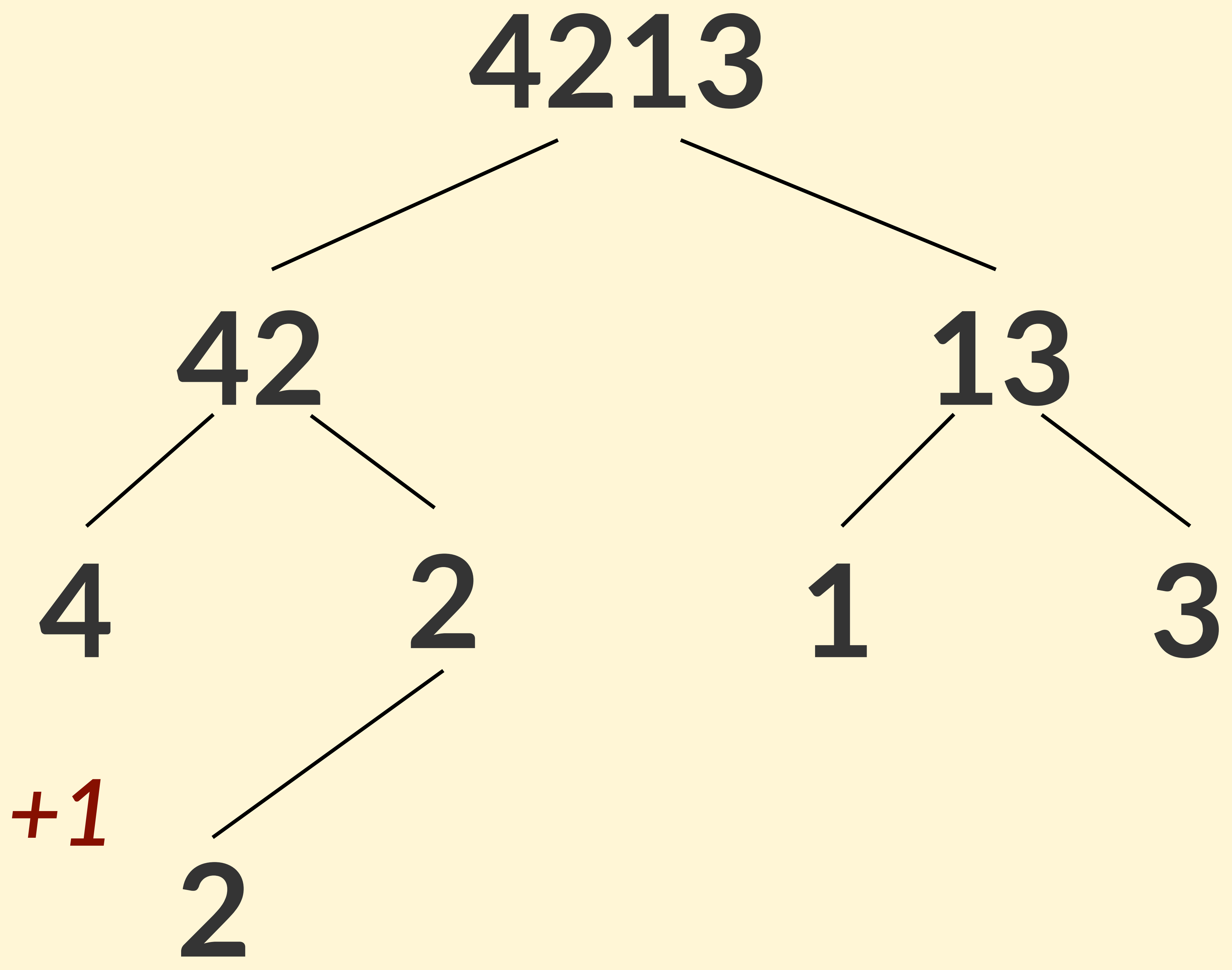
1

3

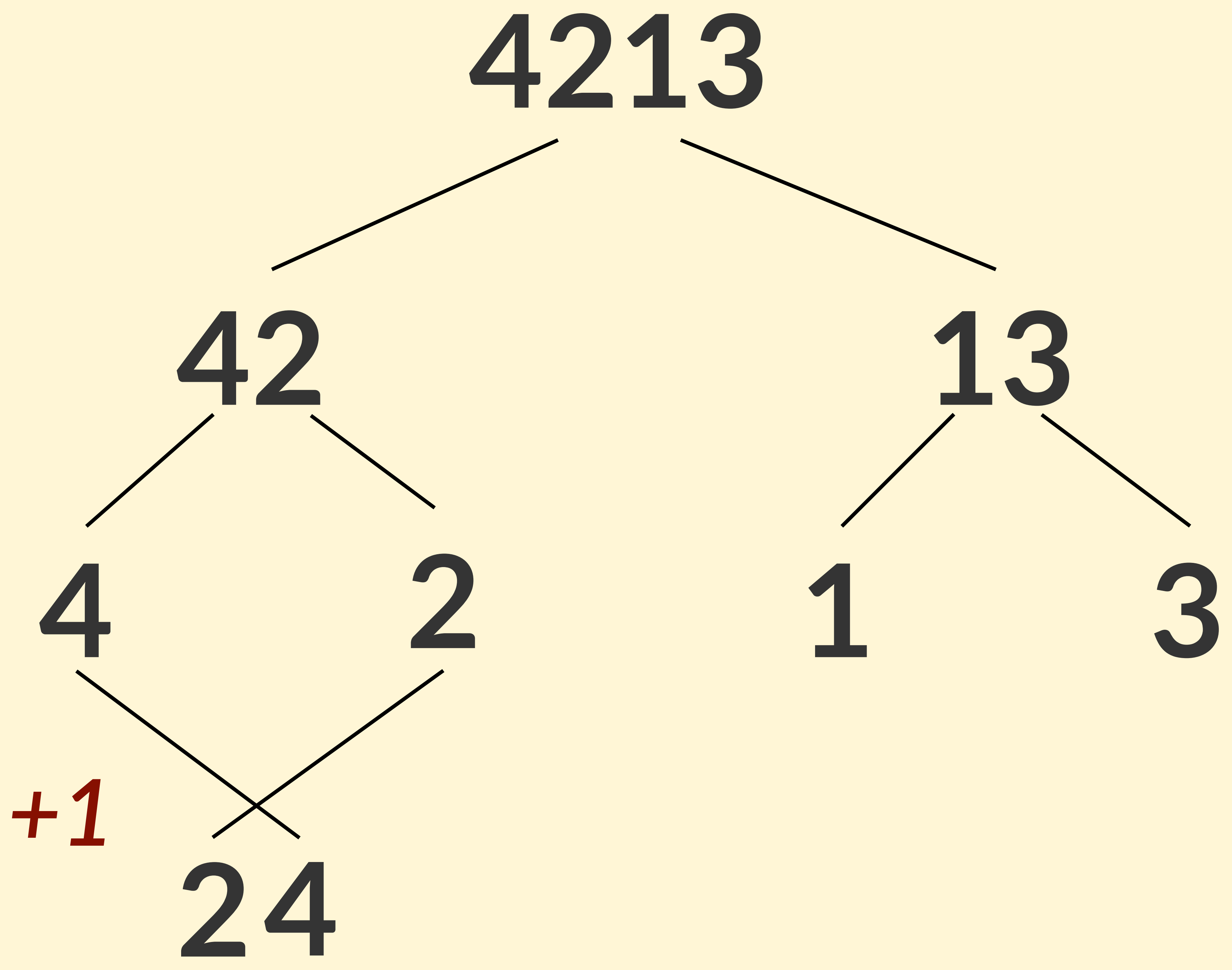
Aside: Finding the Phase



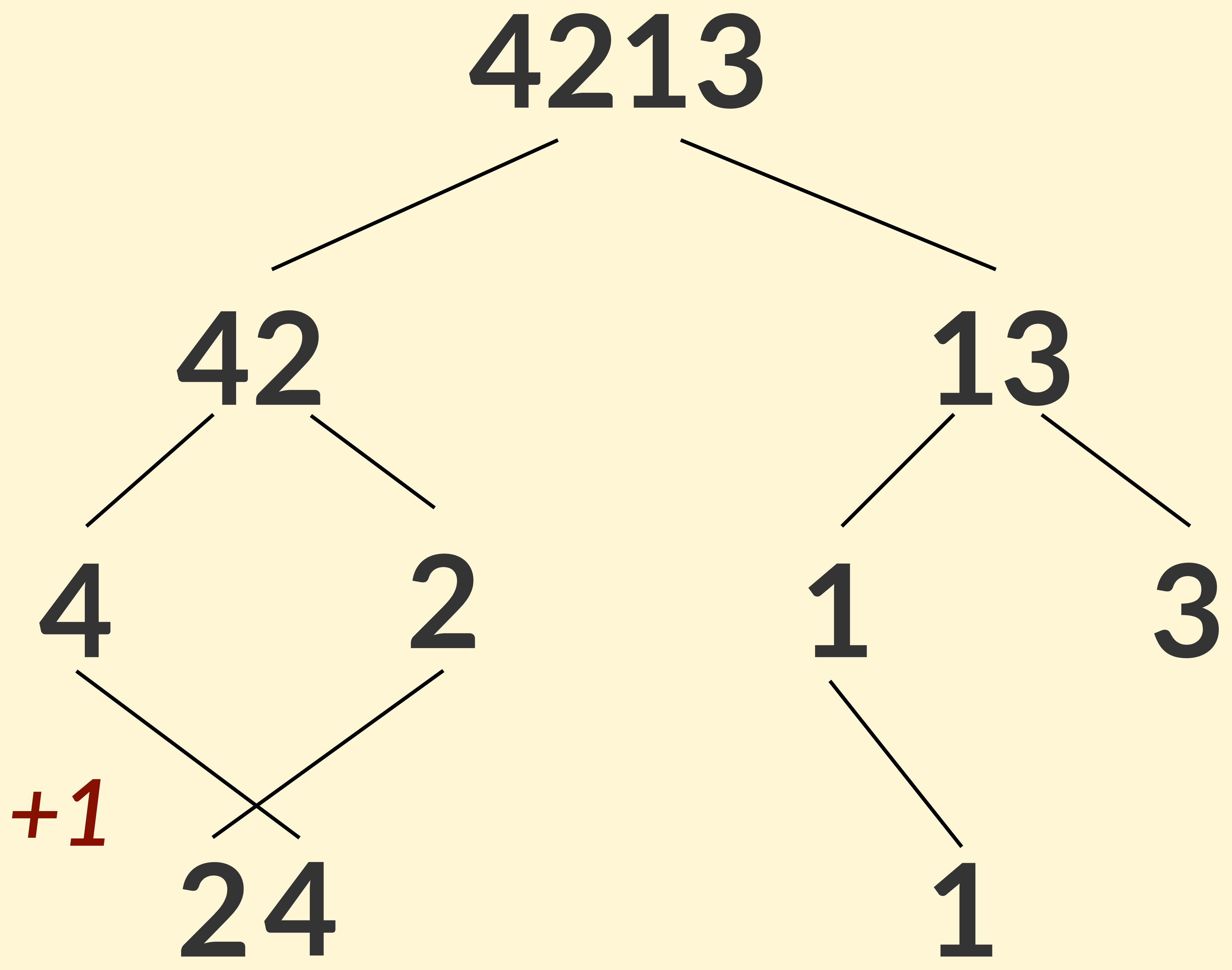
Aside: Finding the Phase



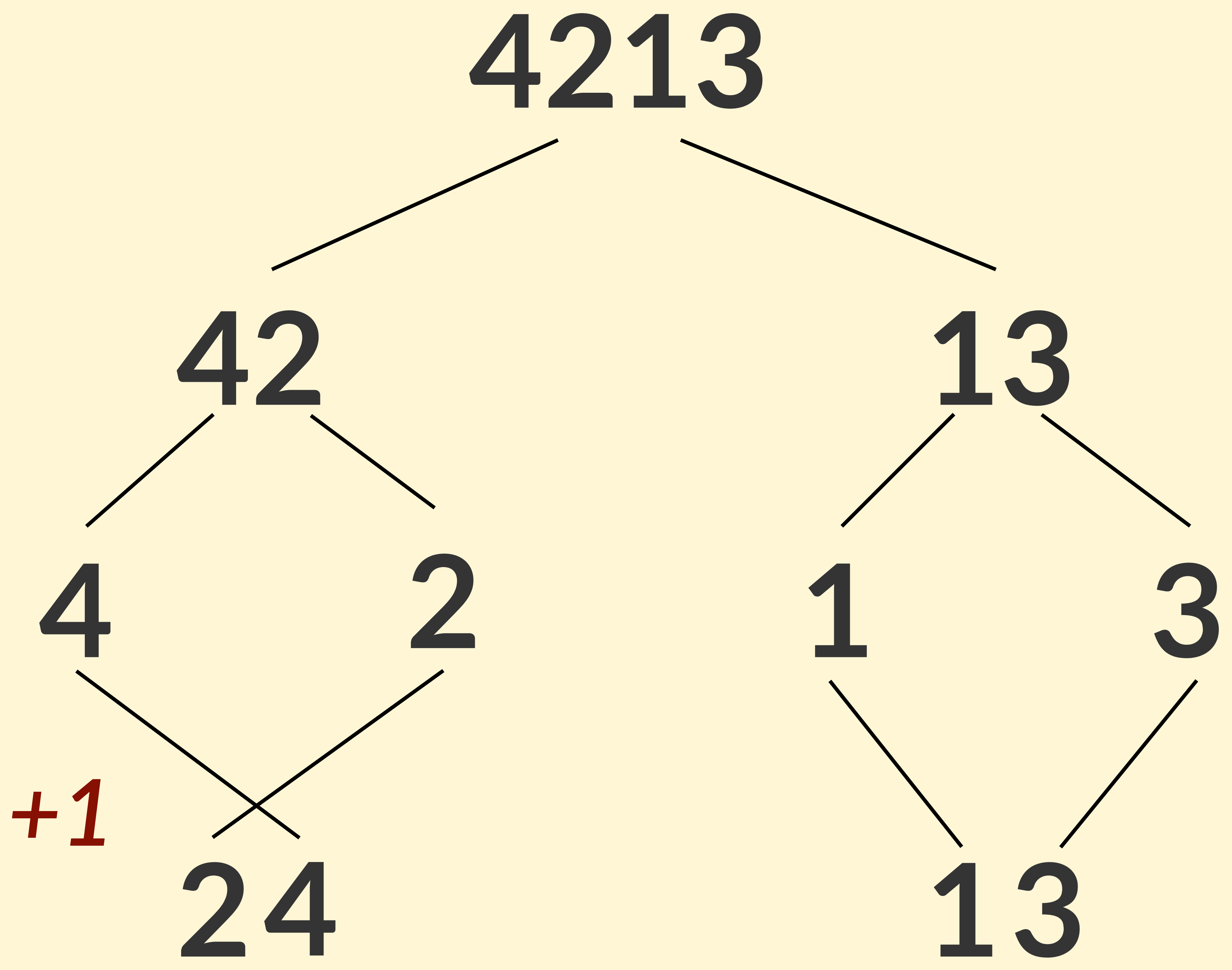
Aside: Finding the Phase



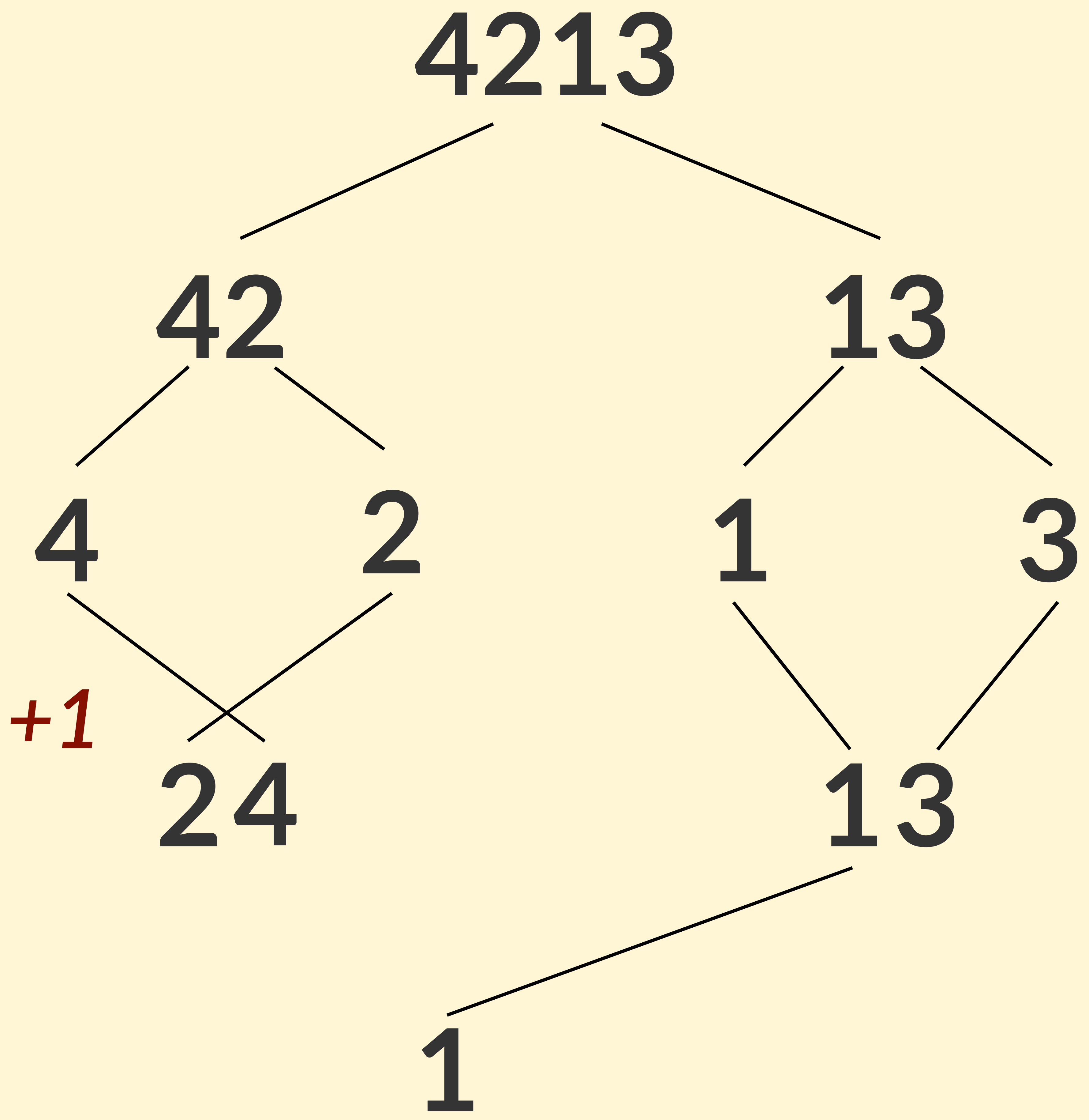
Aside: Finding the Phase



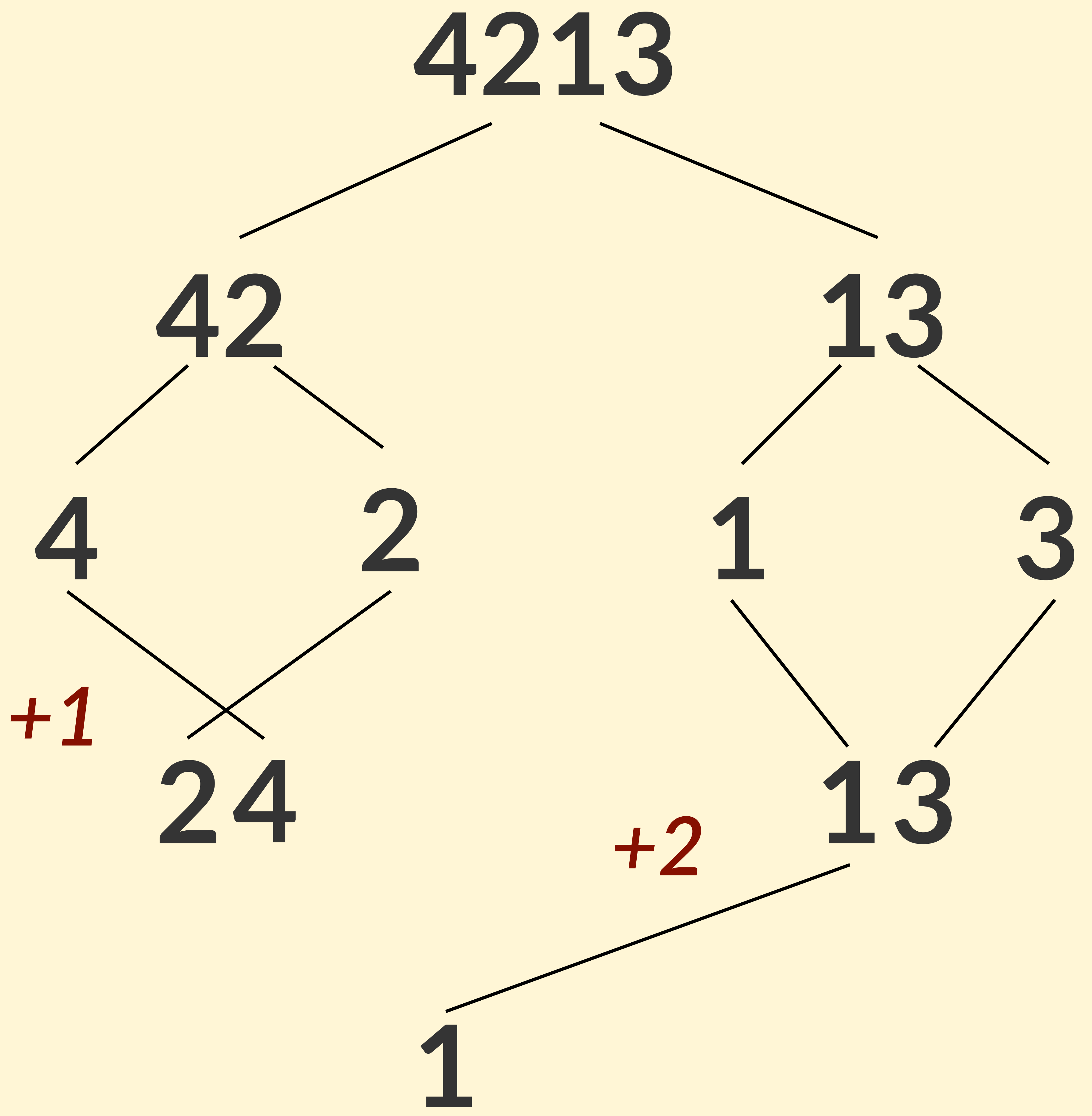
Aside: Finding the Phase



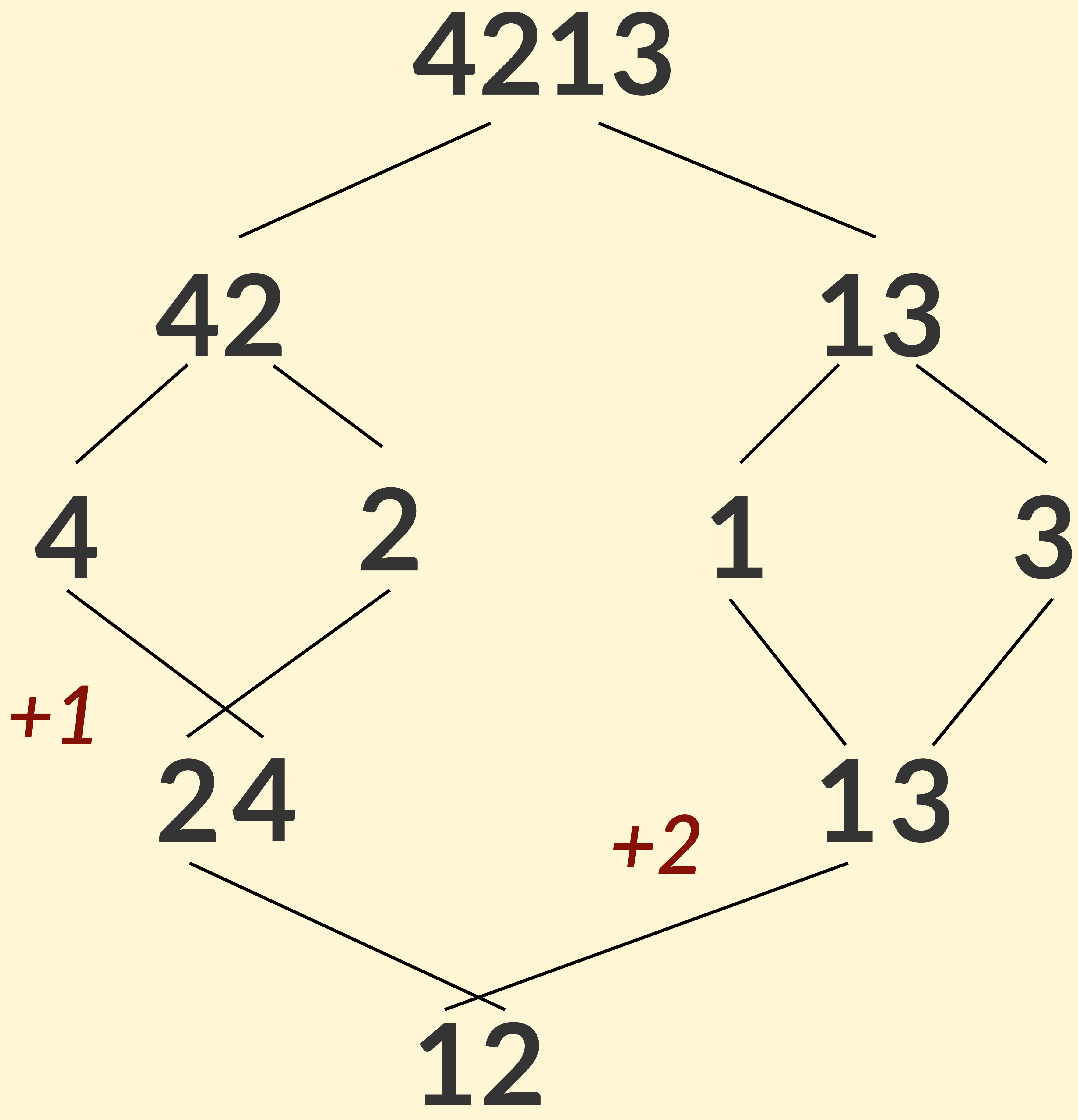
Aside: Finding the Phase



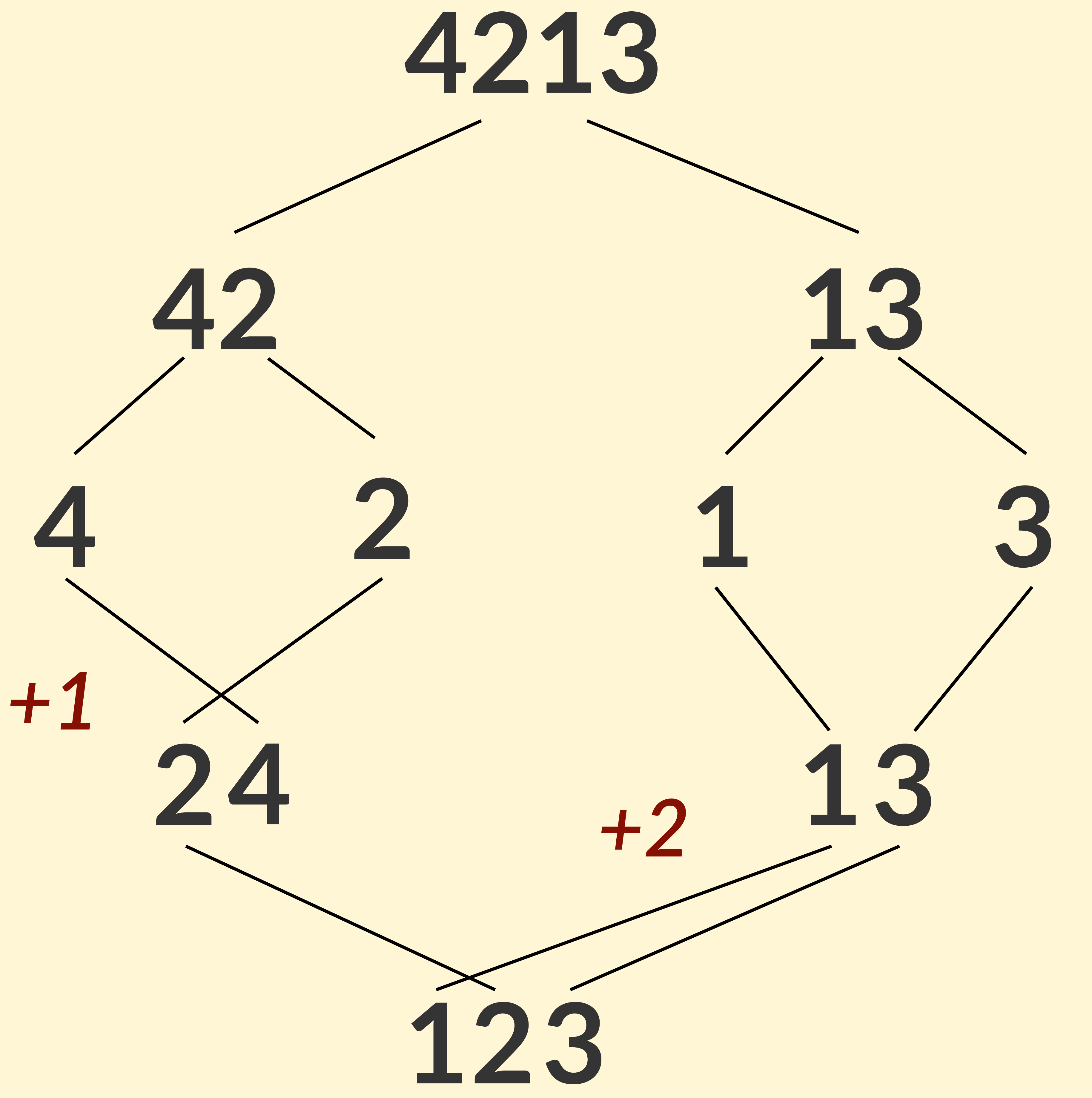
Aside: Finding the Phase



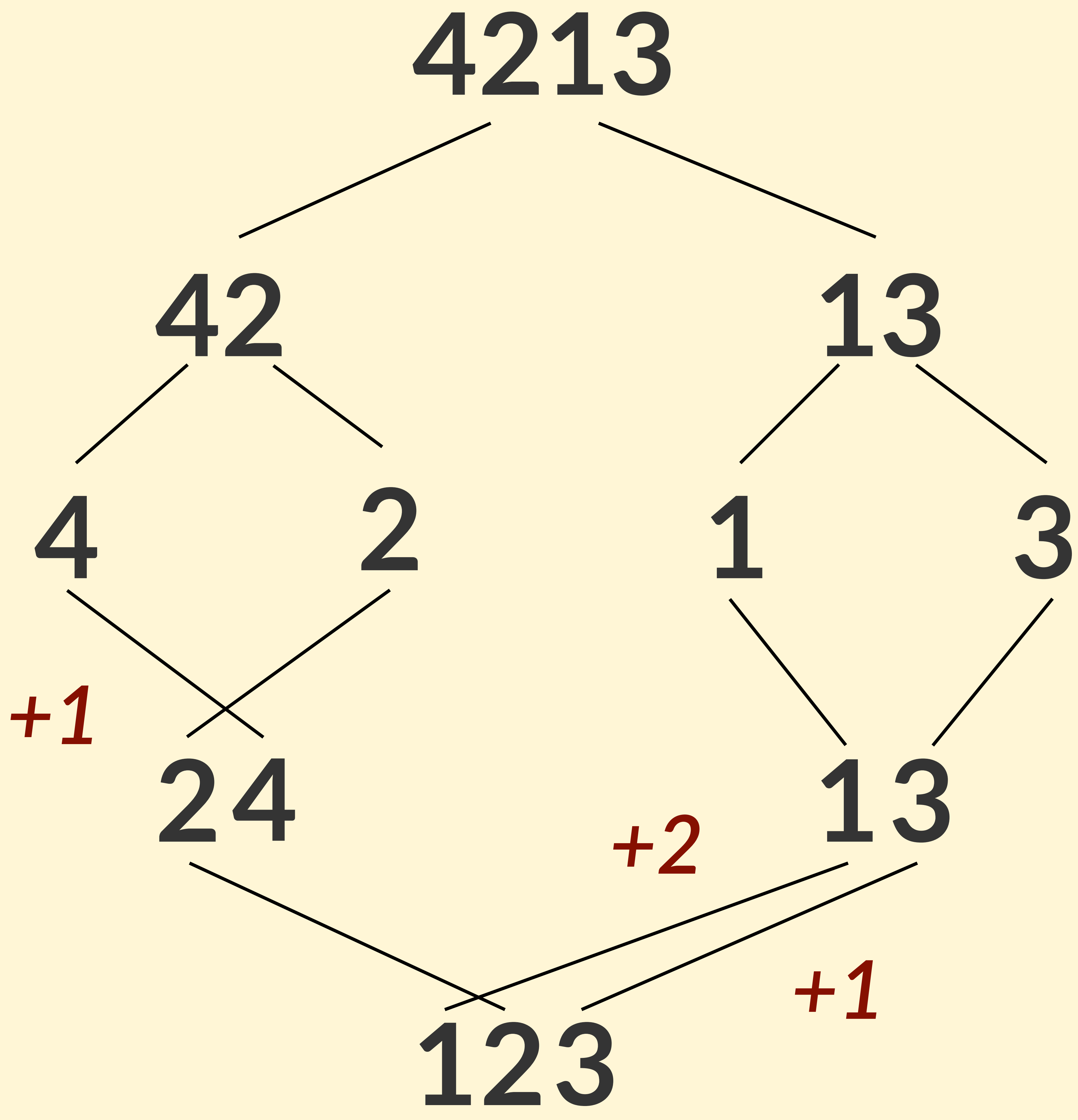
Aside: Finding the Phase



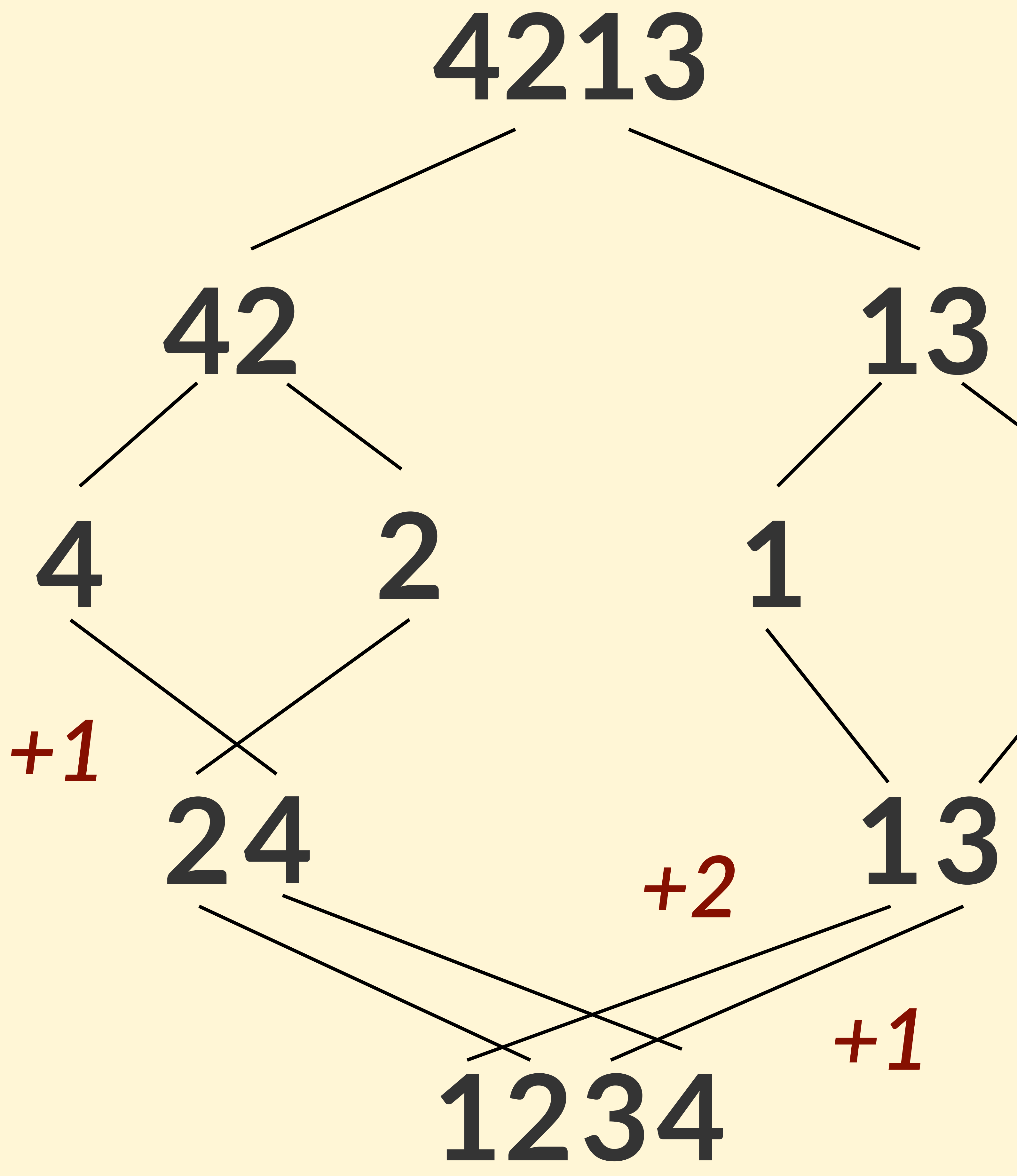
Aside: Finding the Phase



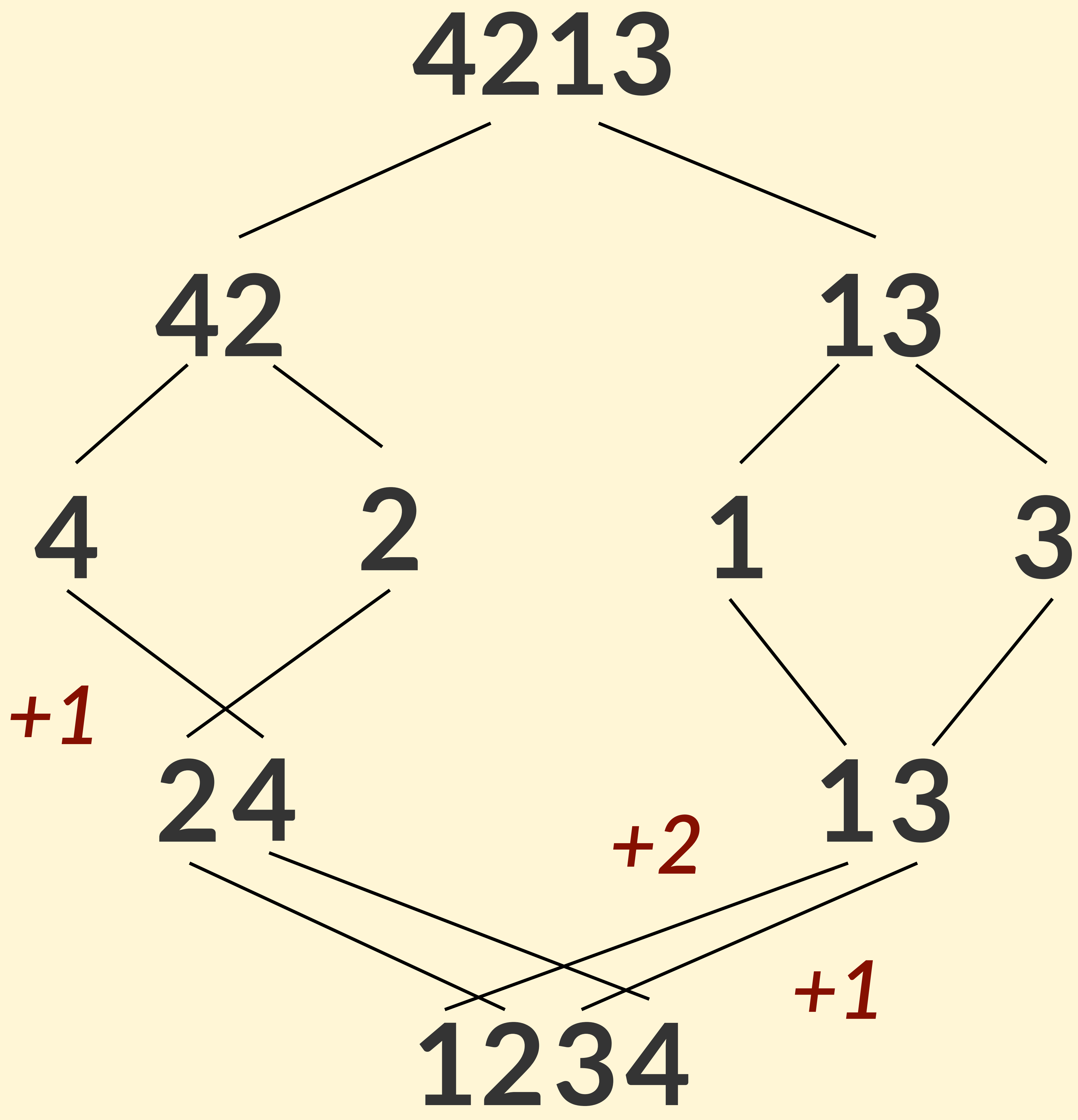
Aside: Finding the Phase



Aside: Finding the Phase

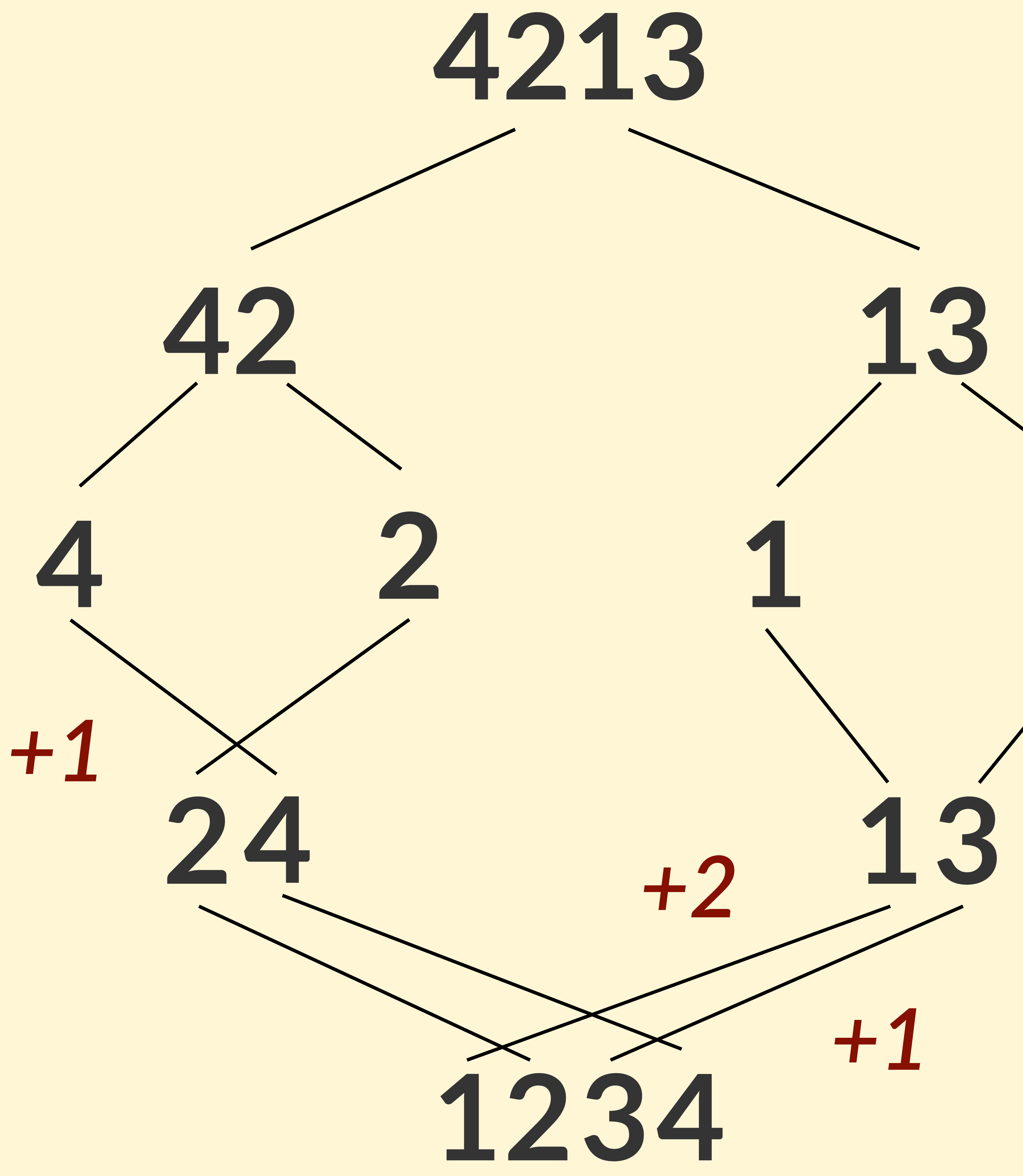


Aside: Finding the Phase



4 Inversions

Aside: Finding the Phase



4 Inversions

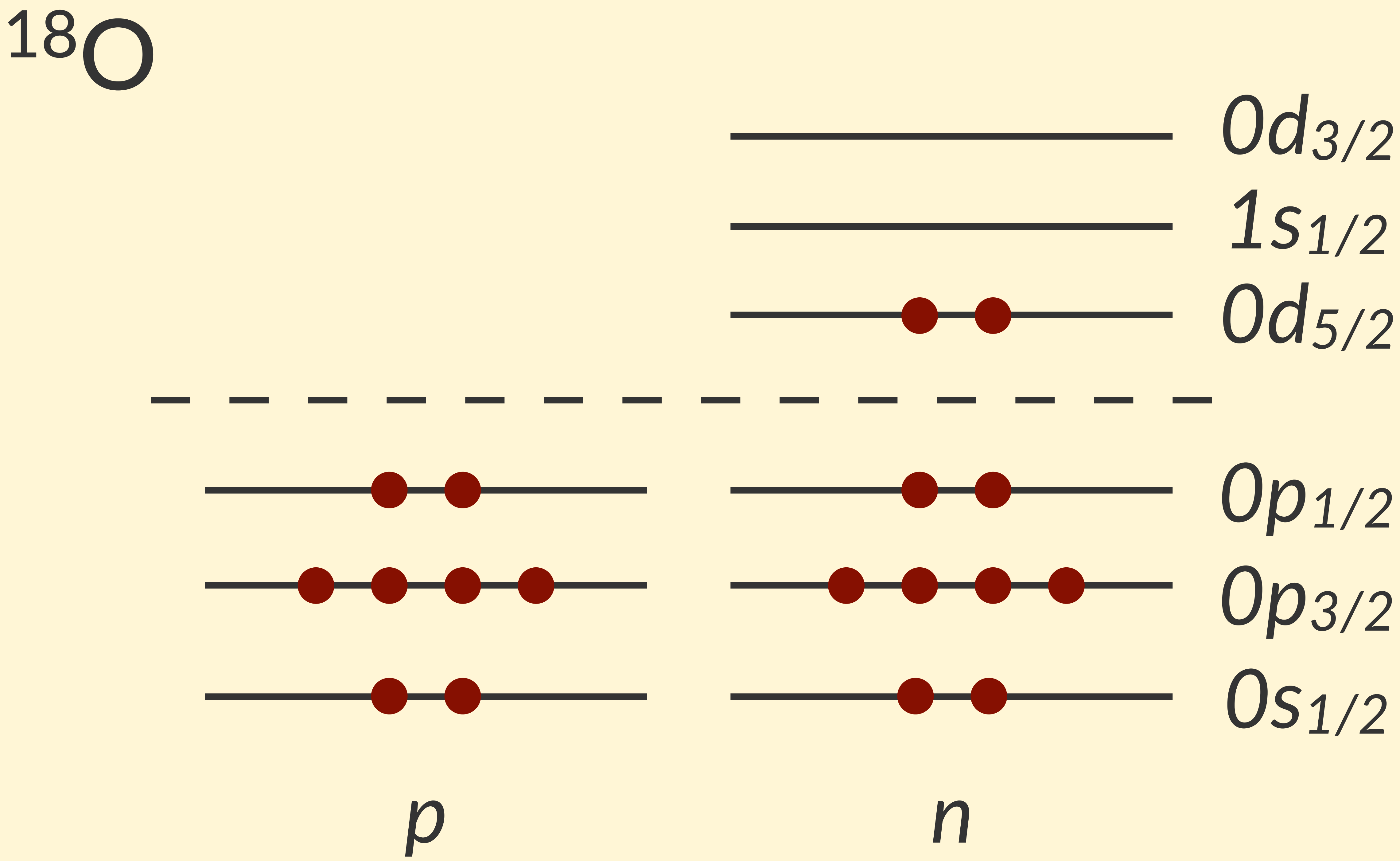
Phase = $(-1)^4$ = +1

Hamiltonian

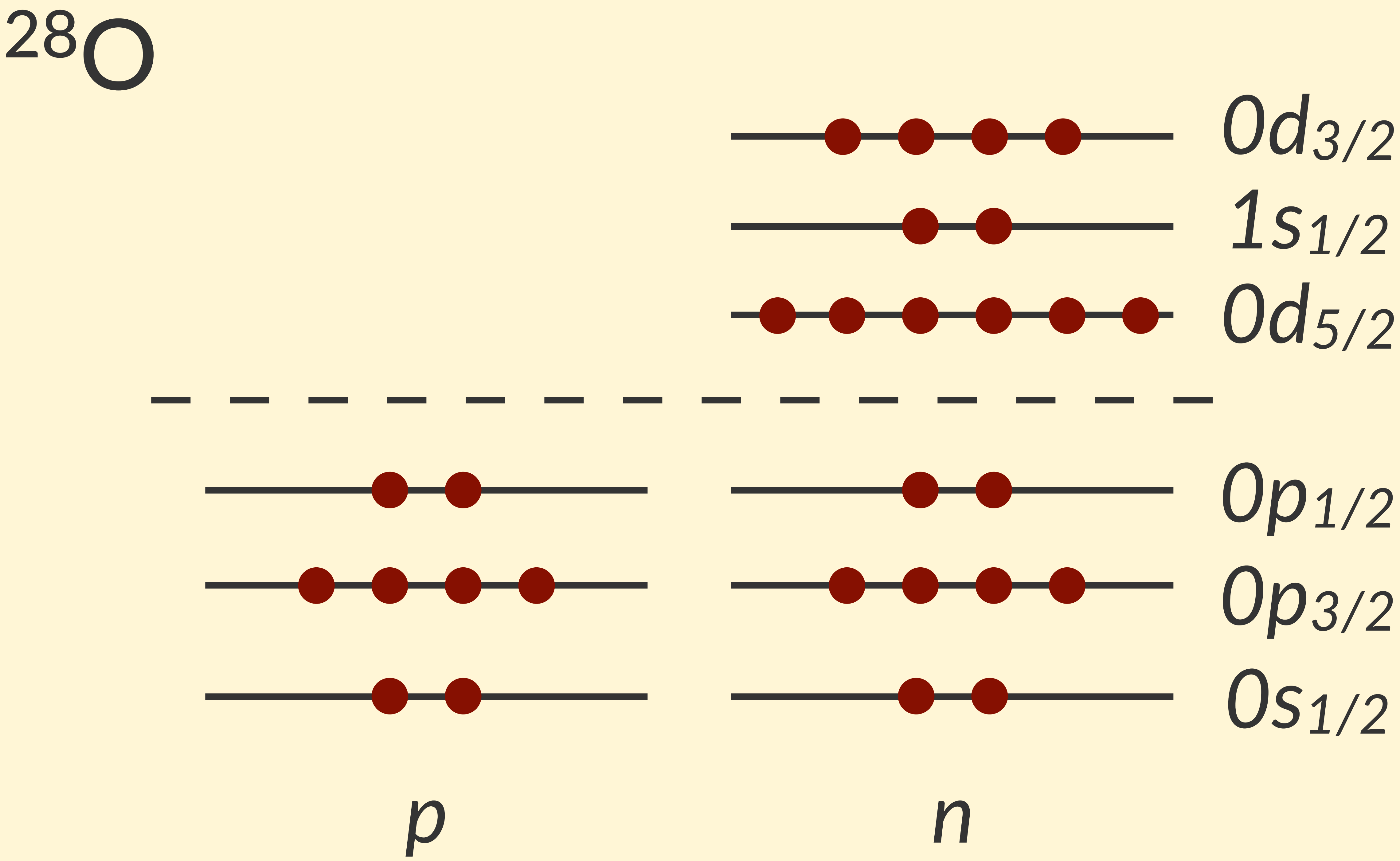
- Iterate over each ket
- Maintain Pauli principle and rules from second quantization
- Find phase using merge sort
- Use matrix elements from USDB file
- Diagonalize with built-in eigenvalue solver from NumPy

```
def find_hamiltonian_matrix(sds, states, interaction):

    let hmat be a square matrix with dim of sds
    fill hmat with 0.0

    for each ket in sds:
        for each p, q, r, s, energy in interaction:
            if r not in ket or s not in ket:
                skip and continue

            if p in ket and (p != r and p != s):
                skip and continue

            if q in ket and (q != r and q != s):
                skip and continue

            copy ket to new_ket
            replace s with q in new_ket
            replace r with p in new_ket
            sort new_ket and find number of inversions
            let i be the matrix row for ket
            let j be the matrix column for new_ket

            hmat[i, j] += energy * (-1)^(number of inversions)
            hmat[j, i] = hmat[i, j]

    for each diagonal element:
        hmat[i, i] += sum(single particle energies)

    return hmat
```

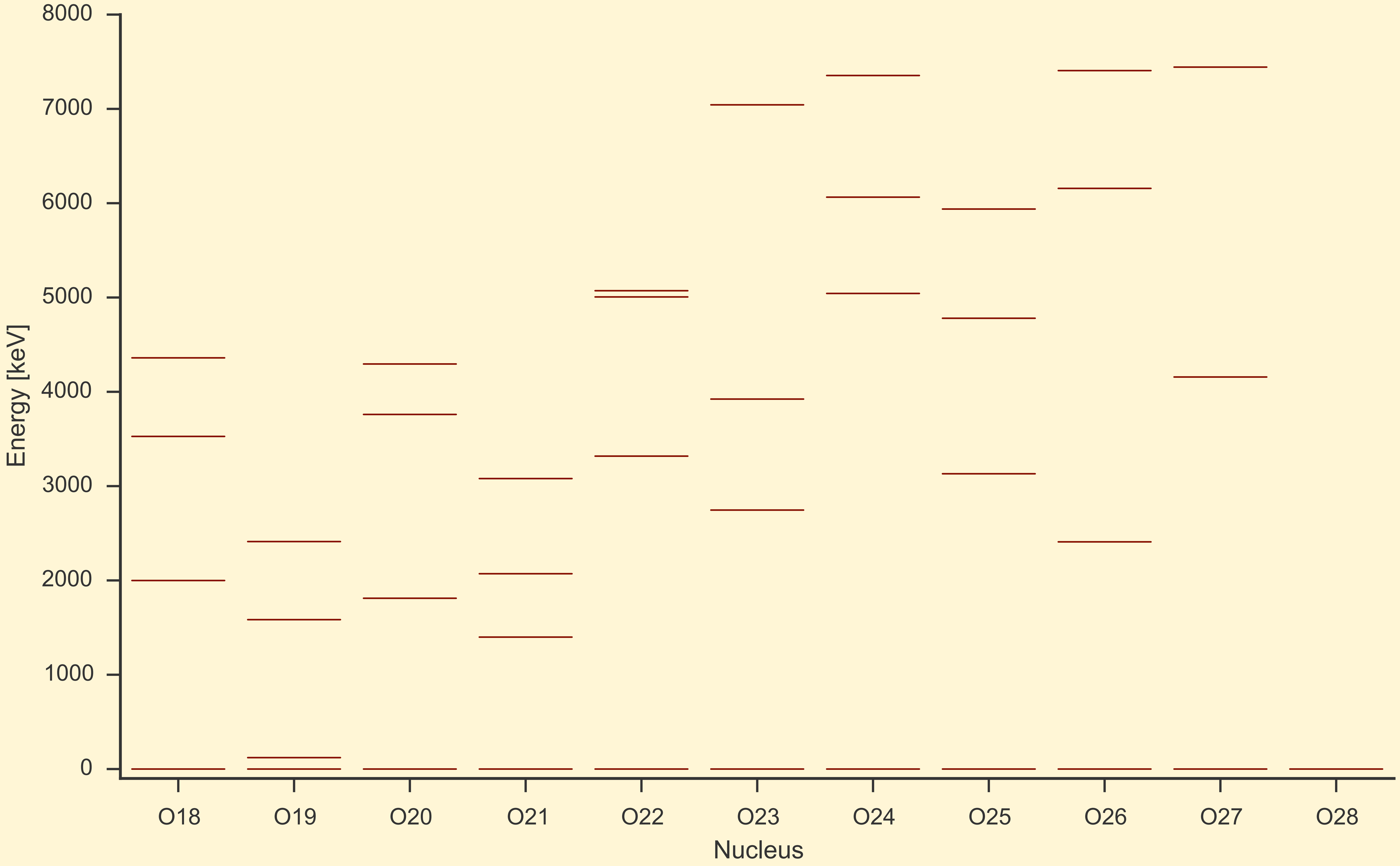
Test: Oxygen isotopes



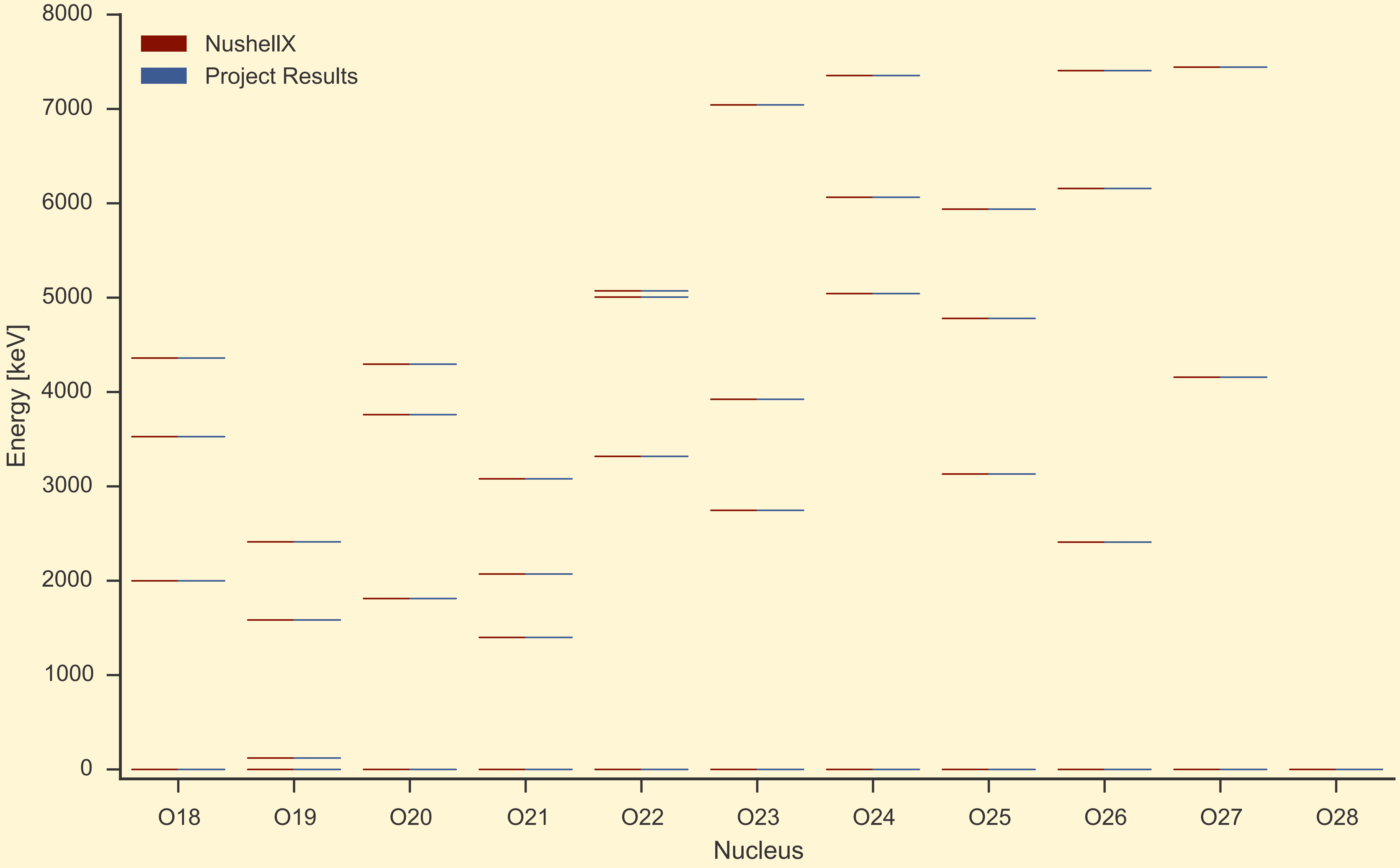
Test: Oxygen isotopes



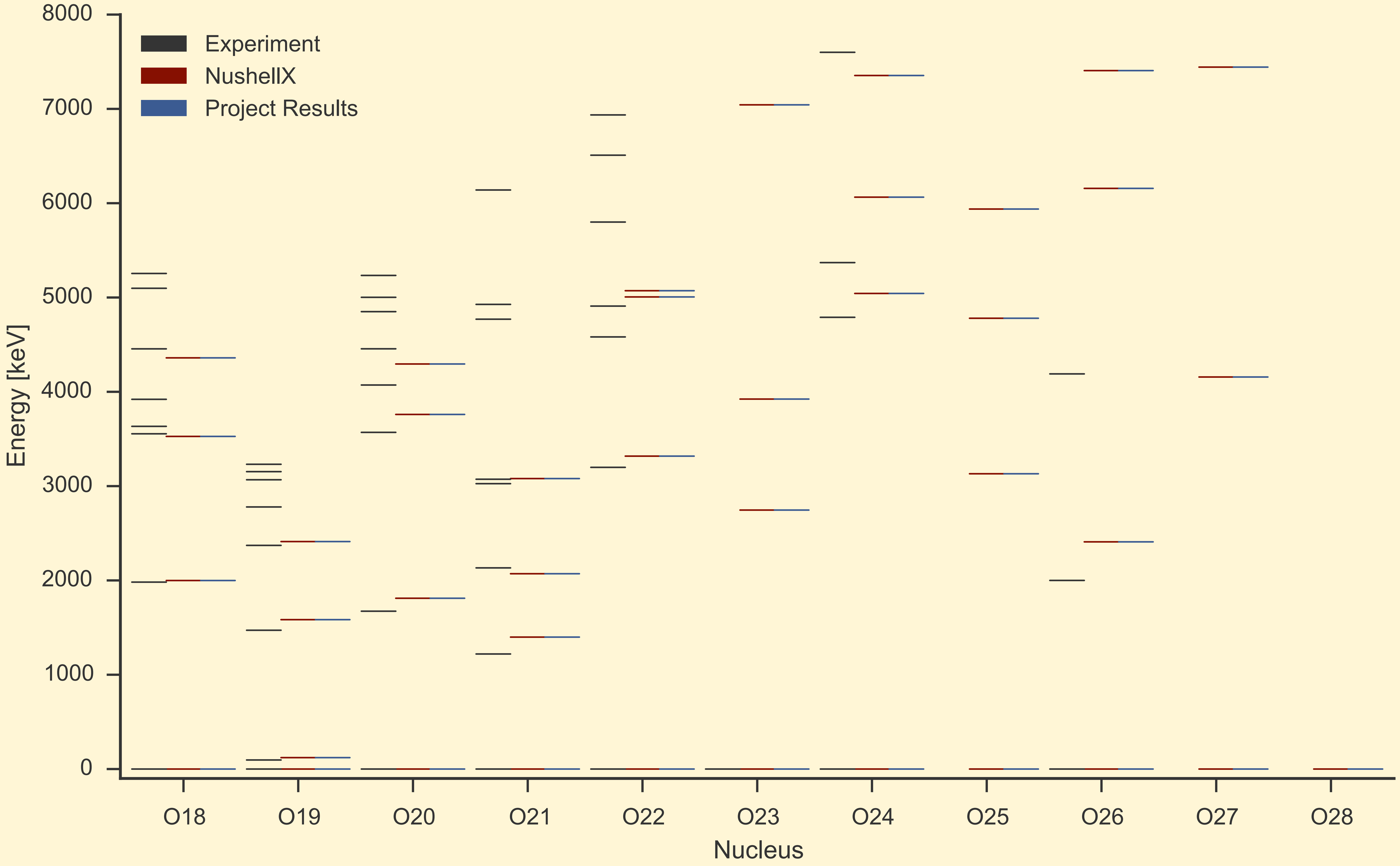
Levels


8000
7000
6000
5000
4000
3000
2000
1000
0
Energy [keV]
O18
O19
O20
O21
O22
O23
O24
O25
O26
O27
O28
Nucleus

Compared to NuShellX



Compared to experiment


Experiment
NushellX
Project Results
Energy [keV]
O18
O19
O20
O21
O22
O23
O24
O25
O26
O27
O28
Nucleus