# Grundy Eigenvalue

We want to find bound for the eigenvalues of the following matrix:

$$S_n = \begin{pmatrix} S_{n-1} & \frac{1}{a}\mathbb{1} \\ \frac{1}{a}\mathbb{1}^T & 0 \end{pmatrix}$$

Where $S_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ and $a = 2^{(n-2)/2}$

The Weyl inequalities states that if we have matrices $A$ and $B$ and $\lambda_k^\downarrow(A)$ corresponds to the $k-$th largest eigenvalue of $A$, then,

$$\lambda_k^\downarrow(A+B) \leq \lambda_i^\downarrow(A) + \lambda_{k-i+1}^\downarrow(B),$$

and therefore,

$$\lambda_{\max}(A+B) \leq \lambda_{\max}(A) + \lambda_{\max}(B),$$

$$\begin{aligned}
\lambda_{\max}(S_n) &\leq \lambda_{\max}(\begin{pmatrix} S_{n-1} & 0 \\ 0 & 0 \end{pmatrix}) + \lambda_{\max}(\begin{pmatrix} 0 & \frac{1}{a}\mathbb{1} \\ \frac{1}{a}\mathbb{1}^T & 0 \end{pmatrix}) \\
&= \lambda_{\max}(S_{n-1}) + \frac{1}{a}\lambda_{\max}(\begin{pmatrix} 0 & \mathbb{1} \\ \mathbb{1}^T & 0 \end{pmatrix}) \\
&= \lambda_{\max}(S_{n-1}) + \frac{\sqrt{n-1}}{a}
\end{aligned}$$

Where in the last inequality we consider the largest eigenvalue of the star graph.

Since we found an upper bound for the difference between the greatest eigenvalue of $S_n$ and $S_{n-1}$, we can derive an upper bound for the eigenvalue of $S_n$:

$$\begin{aligned}
\lambda_{max}(S_n) &= \lambda_{max}(S_2) + \sum_{i=3}^{n} \lambda_{\max}(S_i) - \lambda_{\max}(S_{i-1}) \\
&\leq 1 + \sum_{i=3}^{n} \frac{\sqrt{i-1}}{a} \\
&= 1 + \sum_{i=3}^{n} \sqrt{\frac{i-1}{2^{i-2}}} \\
&< 6.862
\end{aligned}$$

Last inequality checked in wolfram.

```
In [1]: def atom_quotient_matrix_coutinho(n):
            A = matrix(SR, n)
            for i in range(2, n+1):
                for j in range(1, i):
                    A[i-1, j-1] = A[j-1, i-1] = 1/sqrt(2**(i-2));
            return A

In [2]: A = atom_quotient_matrix_coutinho(6)
        print(A)

        [          0           1 1/2*sqrt(2)         1/2 1/4*sqrt(2)         1/4]
        [          1           0 1/2*sqrt(2)         1/2 1/4*sqrt(2)         1/4]
        [1/2*sqrt(2) 1/2*sqrt(2)           0         1/2 1/4*sqrt(2)         1/4]
        [        1/2         1/2         1/2           0 1/4*sqrt(2)         1/4]
        [1/4*sqrt(2) 1/4*sqrt(2) 1/4*sqrt(2) 1/4*sqrt(2)           0         1/4]
        [        1/4         1/4         1/4         1/4         1/4           0]

In [3]: import numpy as np
        A = atom_quotient_matrix_coutinho(100)
        A = matrix(RR, A)
        max(np.linalg.eig(A)[0])

Out[3]: 2.50306089374004
```

Figure 1: Experimental result