```
import sys
import time

import numpy as np
import seaborn as sns
from scipy import stats
import matplotlib.pylab as plt


sys.setrecursionlimit(100000)


class PerfTree(object):
    def __init__(self, m):
        self.m = np.array(m)
        bin_ns = []
        for i in range(self.m.shape[0]):
            bin_ns.append(''.join(self.m[:, i]))
        for i in range(self.m.shape[1]):
            bin_ns_0, bin_ns_1 = [], []
            for bin_n in bin_ns:
                if bin_n[i] == '1':
                    bin_ns_0.append(bin_n)
                else:
                    bin_ns_1.append(bin_n)
            bin_ns = bin_ns_0 + bin_ns_1
        self.m = np.array([list(x) for x in bin_ns]).T.astype(int)
        self.mut_ind = [0] * self.m.shape[1]

    def Size(self):
        return self.m.shape

    def CheckPerf(self, r_i=None, c_i=0):
        if r_i is None:
            r_i = range(self.m.shape[0])

        if c_i == self.m.shape[1]:
            return True

        r_i_0, r_i_1 = [], []
        for r in r_i:
            if self.m[r, c_i] == 0:
                r_i_0.append(r)
            else:
                r_i_1.append(r)

        if len(r_i_1) and self.mut_ind[c_i]:
            return False
        if len(r_i_1):
            self.mut_ind[c_i] = 1

        res1 = self.CheckPerf(r_i_1, c_i + 1) if len(r_i_1) else
True
        if res1 is False:
            return False
        return self.CheckPerf(r_i_0, c_i + 1) if len(r_i_0) else
True


pattern = 'a1data$.txt'
files = []
for i in range(1, 7):
    files.append(pattern.replace('$', str(i)))

sizes, times = [], []
for f in files:
    print(f)
    with open(f, 'r') as f:
        m = [list(x.strip()) for x in f.readlines()]

    starttime = time.process_time()
    tree = PerfTree(m)
    print(tree.CheckPerf())
    endtime = time.process_time()
    sizes.append(tree.Size()[0])
    times.append(endtime - starttime)
    print(sizes[-1], times[-1])

sizes, times = np.array(sizes), np.array(times)
sizes, times = np.log10(sizes), np.log10(times)
ax = sns.regplot(sizes, times)
ax.set(xlabel='log_10 n', ylabel='log_10 f(n, m)',
       title='Checking Perfect Phylogeny speed')
plt.savefig('3_speed.pdf', format='pdf')
slope, intercept, r_value, p_value, std_err =
stats.linregress(sizes, times)
print(slope, intercept, r_value, p_value, std_err)
```