[1]:

```
from itertools import permutations
import numpy as np

def hamiltonian_cycles(cities, symmetric = False):
    'returns a list of all possible hamiltonian cycles for a given list of cities'
    start = cities[0]
    cycles = []
    for permutation in permutations(cities[1:]):
        cycle = start + ''.join(permutation) + start
        if symmetric:
            if cycle[::-1] not in cycles:
                cycles.append(cycle)
        else:
            cycles.append(cycle)
    return cycles

def map_indices(cities, symmetric = False):
    'returns all the hamiltonian cycles and the indices'
    cycles = hamiltonian_cycles(cities, symmetric = symmetric)
    index_map = {cities[i]: str(range(len(cities))[i]) for i in range(len(cities))}
    indices_cycles = []
    for cycle in cycles:
        indices_city = ''
        for city in cycle[1:]:
            indices_city += index_map[city]
        indices_cycles.append(indices_city)
    return cycles, indices_cycles

def sort_indices(cycles, indices_cycles):
    'sorts the indices'
    results = []
    for cycle, index in zip(cycles, indices_cycles):
        pairs = list(zip(cycle, index))
        sorted_pairs = sorted(pairs, key = lambda pair: pair[0]) # sorts by city name
        sorted_index = ''.join([pair[1] for pair in sorted_pairs])
        results.append([cycle, index, sorted_index])
    return results
```

[2]:

```
cities = ['A', 'B', 'C', 'D', 'E']
cycles, indices_cycles = map_indices(cities, symmetric = True)
cycles, indices_cycles
```

[2]:

```
(['ABCDEA',
  'ABCEDA',
  'ABDCEA',
  'ABDECA',
  'ABECDA',
  'ABEDCA',
  'ACBDEA',
  'ACBEDA',
  'ACDBEA',
  'ACEBDA',
  'ADBCEA',
  'ADCBEA'],
 ['12340',
  '12430',
  '13240',
  '13420',
  '14230',
  '14320',
  '21340',
  '21430',
  '23140',
  '24130',
  '31240',
  '32140'])
```

[3]:
```
table = sort_indices(cycles, indices_cycles)
print("cycle, index, sorted_index")
table
```

```
cycle, index, sorted_index
```

[3]:
```
[['ABCDEA', '12340', '12340'],
 ['ABCEDA', '12430', '12403'],
 ['ABDCEA', '13240', '13420'],
 ['ABDECA', '13420', '13042'],
 ['ABECDA', '14230', '14302'],
 ['ABEDCA', '14320', '14023'],
 ['ACBDEA', '21340', '23140'],
 ['ACBEDA', '21430', '24103'],
 ['ACDBEA', '23140', '24310'],
 ['ACEBDA', '24130', '23401'],
 ['ADBCEA', '31240', '32410'],
 ['ADCBEA', '32140', '34120']]
```

[4]:
```
# Base 10 and 2 conversions
table = np.array(table)
indices = table[:,2]
base_10 = np.array([int(indices[i], len(cities)) for i in range(len(indices))])
base_2  = np.array(["{0:b}".format(base_10[i]) for i in range(len(base_10))])
table = np.append(table, base_10.reshape(-1,1), axis=1)
table = np.append(table, base_2.reshape(-1,1), axis=1)
print("cycle, index, sorted_index, base 10, base 2 \n",table)
```

```
cycle, index, sorted_index, base 10, base 2
 [['ABCDEA' '12340' '12340' '970' '1111001010']
 ['ABCEDA' '12430' '12403' '978' '1111010010']
 ['ABDCEA' '13240' '13420' '1110' '10001010110']
 ['ABDECA' '13420' '13042' '1022' '1111111110']
 ['ABECDA' '14230' '14302' '1202' '10010110010']
 ['ABEDCA' '14320' '14023' '1138' '10001110010']
 ['ACBDEA' '21340' '23140' '1670' '11010000110']
 ['ACBEDA' '21430' '24103' '1778' '11011110010']
 ['ACDBEA' '23140' '24310' '1830' '11100100110']
 ['ACEBDA' '24130' '23401' '1726' '11010111110']
 ['ADBCEA' '31240' '32410' '2230' '100010110110']
['ADCBEA' '32140' '34120' '2410' '100101101010']]
```