David Luo
COS 488

5/5

Compute the percentage of permutations having no singleton or doubleton cycles and compare with the asymptotic estimate from analytic combinatorics, for $N = 10$ and $N = 20$.

Our lecture slides show that the number of permutations of length N without any cycles of length M or less is roughly $N!/e^{H_M}$, so in our case the percentage of permutations having no singleton or doubleton cycles should be $100(e^{-H_2})$, or 22.3%. Using Python 2.7 and a sample size of one million, I found that the proportion of size 10 and 20 permutations with no singleton or doubleton cycles was indeed right around 22.3%.

My code is shown below, along with the output.

```
#!/usr/bin/python
import numpy, math

# constants
sampleSize = 1000000
N1 = 10
N2 = 20

# returns the inverse of a permutation p
def inverse(p):
	q = [0] * len(p)
	for i, v in enumerate(p):
		q[v] = i
	return q

# returns true if list p has no singleton or doubleton cycles, false if so
def hasNoSDCycles(p):
	q = inverse(p)
	for i in xrange(len(p)):
		if (q[i] == p[i]):
			return False
	return True
```

```
# calculate number of permutations without singleton or doubleton cycles
def main():
	# N = 10
	count1 = 0
	for i in xrange(sampleSize):
		p = numpy.random.permutation(N1).tolist()
		if (hasNoSDCycles(p)):
			count1 += 1
	print count1
	# N = 20
	count2 = 0
	for i in xrange(sampleSize):
		p = numpy.random.permutation(N2).tolist()
		if (hasNoSDCycles(p)):
			count2 += 1
	print count2
main()

# Output below:
# >python IV.1.python
# 222987
# 223212
```