CATAM Part II - 17.3 - Hamiltonian Cycles

Introduction

Our first task is to create a sensible data structure to store our graph in, and also write a function that generates a random graph from $\mathcal{G}(n,p)$. Since I'm coding in python, I've implemented a Graph class, which stores all our graph's information, and will contain various methods pertaining to our graph. It allows for us to generate a random graph or input a vertex and edge set.

Question 1

The simplest, and least efficient way to check if a graph has a Hamiltonian cycle is to check each of the $n!$ individual permutations of the vertex set. We can reduce checking slightly since cyclic permutations are equivalent, sow we only have $(n-1)!$. The hard part here is iterating over all permutations, since theres eventually too many to store in memory. We can overcome this by, say, Heap's Algorithm[1], but our implementation uses the *permutations* iterator from the standard python library *itertools*. It can be found under $naive_hamiltonian()$ as a method of the Graph class. The following data is gathered by $q1()$. TODO maybe implement own iterator

```
* Order=3, Size=3, {1: {2, 3}, 2: {1, 3}, 3: {1, 2}} * has a hamiltonian cycle:
  [1,2,3]
* Order=3, Size=2, {1: {2}, 2: {1, 3}, 3: {2}} * has no hamiltonian cycle
* Order=4, Size=3, {1: {2, 3}, 2: {1, 3}, 3: {1, 2}} * has no hamiltonian cycle
```

p/n	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**
0.1	985	1075	978	1009	938	1059	1033	975	1020
0.3	3036	3019	2966	3048	3029	2954	2968	2968	3075
0.5	4911	5022	4933	5077	4997	5026	4894	4953	5026
0.7	6916	6978	7083	6996	7009	6994	7009	6983	7088
0.9	9025	8990	9000	8954	8982	9021	8959	9040	9025

Table 1: Number of graphs containing Hamiltonian cycles from a selection of 10000 taken from $\mathcal{G}(n,p)$

a/n	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**
0.1	332	293	262	246	216	200	168	164	150
0.55	1838	1673	1494	1274	1158	1087	910	887	828
1	3484	2998	2509	2216	2149	1912	1715	1625	1544
1.45	4925	4330	3734	3305	3148	2730	2512	2366	2195
1.9	6583	5685	4924	4406	4042	3646	3385	3067	2871

Table 2: Number of graphs containing Hamiltonian cycles from a selection of 10000 taken from $\mathcal{G}(n, a\log n/n)$

Question 2

Suppose we have n vertices. The worst case is we have no Hamiltonian cycle, so each of the $(n-1)!$ permutations is checked. We assume each simple operation, ie iterating and looking up a value in a list take time c. For each. At worst, we have the first $n-1$ entries of our permutation form a Hamiltonian path [2], so we do $2n$ lookups and n iterations, giving us a time of $3nc$ per permutation. Supposing there is a time d taken to iterate our permutation, we end up with a total running time of $(n-1)!(3nc+d) = O(n!)$.

Now for an average case, supposing an average graph can be taken from $\mathcal{G}(n,1/2)$. Supposing such an average graph has k Hamiltonian cycles, from Table 1 it seems reasonable to assume $k = 1/2$. So in the case we have a Hamiltonian cycle, we expect to find it in $(n-1)!/2$ permutations, with each prior permutation failing in $n/2$ steps. This gives a running time of $(n-1)!(3nc+d)/4$. In the other case where we don't have a Hamiltonian cycle, we similarly get $(n-1)!(3nc+d)/2$, so expect a running time of $3(n-1)!(3nc+d)/4 = O(n!)$.

So the algorithm cannot handle too large an n.

[1] `https://en.wikipedia.org/wiki/Heap%27s_algorithm`

[2] This can't happen for each but its an ok approximation

Question 3

Modifying the code slightly, to print the number of graphs G with $\delta(G) < 2$, we find most, at least for $a < 1.45$ for a defined in Table 2, graphs fail to be Hamiltonian because $\delta(G) < 2$. The second range may well have been chosen because of the following theorem, proved in IID Graph Theory.

Theorem. *Let $\omega(n) \to \infty$. If $p = \frac{\log(n)-\omega(n)}{n}$ then G has isolated vertices almost surely. If $p = \frac{\log(n)+\omega(n)}{n}$ then G has no isolated vertices almost surely.*

Proof. Note if $X = \Sigma_A I_A$ is a sum of indicator functions then $\mathrm{Var}(X) = \Sigma_{A,B}\mathbb{P}(A)[\mathbb{P}(B \mid A) - \mathbb{P}(B)]$ simply by expanding.

Now let X be the number of isolated vertices $X = \Sigma_v I_v$ where I_v indicated v being isolated. So

$$\begin{aligned}
\mathrm{Var}(X) &= \Sigma_{u,v}\mathbb{P}(u \text{ isolated})[\mathbb{P}(vu \text{ isolated} \mid uu \text{ isolated}) - \mathbb{P}(v \text{ isolated})] \\
&= (1-p)^{n-1}[1-(1-p)^{n-1}] + n(n-1)(1-p)^{n-1}[(1-p)^{n-2} - (1-p)^{n-1}] \\
&\leq \mathbb{E}(X) + n^2(1-p)^{n-1}(1-p)^{n-2} \\
&= \mathbb{E}(X) + \frac{p}{1-p}(\mathbb{E}(X))^2
\end{aligned}$$

where the first term comes from u and v being the same, and the second term otherwise. Now if $p = \frac{\log(n)+\omega(n)}{n}$

$$\mathbb{E}(X) = \frac{1}{1-p}n(1-p)^n \leq \frac{1}{1-p}ne^{-pn} \to 0$$

So $X = 0$ a.s. by Markov's Inequality. If $p = \frac{\log(n)-\omega(n)}{n}$

$$\mathbb{E}(X) \approx \frac{1}{1-p}ne^{-pn} \to \infty$$

So

$$\frac{\mathrm{Var}(X)}{(\mathbb{E}(X))^2} \leq \frac{1}{\mathbb{E}(X)} + \frac{p}{1-p} \to 0$$

So $X \neq 0$ a.s. by Chebyshev's Inequality.

□

Here we're considering a range of values $p \in [\frac{\log(n)-0.9\log(n)}{n}, \frac{\log(n)+0.9\log(n)}{n}]$, so as n grows, the probability of having isolated vertices for small a grows tends to 1, which agrees with our results.