CS 206: Homework 4 - BagelBot: World Enough, and Time 16:198:206

Problem 1: To Absent Friends

As the CEO-pro-tempore of MagusCorp, newly joined member of the International Bagel Association, and leader of the newly formed World Security Task Force Against Future Threaths, BagelBot doesn't have a lot of time for its bagel duties. As such, BagelBot has implemented the following system - employees at MagusCorp can submit their bagel orders for the next morning, and over night, a fleet of modified DeliveryBots will assemble the bagels, leaving them presented and ready to be picked up the next morning, clearly labeled for ease of picking up.

However, BagelBot notices that every so often, the first person to pick up their bagel will just take one at random from the counter - perhaps distracted, frustrated due to morning traffic on the electrolane, or perhaps succumbing to alien spore infection. As other people arrive, if their bagel is still there, they take it - but if their bagel was taken already, they too just grab at random from the remainder, potentially propagating the problem forward. BagelBot wonders, as this progresses, how likely is it that the last person to arrive gets their bagel?

Let $B(N)$ be the probability that on a morning with N orders, where person 1 grabs a bagel at random, the last person to arrive is able to collect their own bagel.

1) Show that $B(1) = 1$, and $B(2) = 1/2$.
2) Show that for $N \geq 2$,

$$B(N) = \frac{1}{N} \sum_{i=1}^{N-1} B(i) \tag{1}$$

> Hint: $B(N)$ is a probability. You know our methods, Watson. Let all of them be applied to this inquiry.

> Hint 2: Given that person 1 takes person k's bagel - how can you model what happens next?

3) Solve for $B(N)$ in terms of only N, with no recurrent terms. What is $B(1000)$?

To help alleviate this, BagelBot decides that every morning it will put out a special bagel that no one ordered. Anyone randomly grabbing a bagel might take it and not interfere with anyone else's order. This is named Max's Bagel. Let $B_{\max}(N)$ be the probability the last of N people get their bagel when Max's Bagel is added to the queue.

4) What is $B_{\max}(1)$? $B_{\max}(2)$?
5) Solve for $B_{\max}(N)$.

> **Bonus:** Suppose that the bagels are numbered with order numbers, and people arrive to get their bagel in the correct order. This is nice, because people can just grab the first bagel they see, and leave. But suppose that the employees who grab bagels at random grab either the first or second bagel currently in line (rather than from any of the bagels available). Solve for $B(N)$ here. Solve for $B_{\max}(N)$ here - assuming that Max's Bagel is first in line, and then assuming that Max's Bagel is last in line.

Problem 2: Security Concerns

Worried about potential spore infections going undetected, BagelBot wants to position DeliveryBots with spore detectors throughout the building. But there is a lot going on at MagusCorp these days, and BagelBot worries about devoting too many resources to this project, and not enough to other things like preparing the morning bagels, or maintaining the Vancouver Peace Accord. How many bots does BagelBot need to allocate to spore monitoring?

Model the MagusCorp building as a graph, where each vertex represents a location (a room, an intersection of two hallways, a stairwell, the lobby, etc), and each edge between two vertices represents a way to move from one location to the other (for instance, a hallway). Assume that a monitor bot positioned at a vertex can monitor its location, as well as any neighbor of its current location.

In an undirected graph $G = (V, E)$, an **independent set** I is a subset of the vertices such that no two vertices in I are connected by an edge. A **maximal independent set** I is an independent set such that adding any of the remaining vertices of V to I would violate independence - i.e., I cannot be increased.

1) Show that if BagelBot positions MonitorBots on the vertices of a maximal independent set, no bot will be monitoring a location where another bot is stationed, and *every location in the building will be monitored.*

Note however that a maximal independent set is not necessarily the *largest* independent set in G.

2) Give an example of a graph with at least two independent sets, one larger than the other, both maximal.

Let $\alpha(G)$ denote the size of the largest maximal independent set in G.

3) What is $\alpha(G)$ for a complete graph on n vertices? What if G is a cycle on n vertices?

Consider the following greedy algorithm for generating maximal independent sets: starting with an empty set I, process the vertices in V one at a time, adding v to I is v is not connected to any vertex already in I.

4) Argue that the output I of this algorithm is a maximal independent set.
5) Construct an example of a graph on n vertices such that running this greedy algorithm on the vertices in one order yields an independent set of size 1, and processing the vertices in a different order yields an independent set of size $n - 1$, which is maximal.

One way of trying to avoid this dependence on ordering is the use of *randomized algorithms.* Essentially, by processing the vertices in a random order, you can potentially avoid (with high probability) any particularly bad orderings. So consider the following randomized algorithm for constructing independent sets:

- First, starting with an empty set I, add each vertex of G to I independently with probability p.
- Next, for any edges with both vertices in I, delete one of the two vertices from I (at random).
- *Note - in this second step, deleting one vertex from I may remove multiple edges from I!*
- Return the final set I.

4) Argue that the output of this algorithm is an independent set. Is it a maximal independent set?

5) Argue that the expected number of vertices in I after Step 1 is $p|V|$.

6) Argue that the expected number of *edges* in I after Step 1 is $p^2|E|$.

7) Argue that the expected size of I after Step 2 is ***greater than or equal to*** $p|V| - p^2|E|$.

8) What value of p produces the largest expected independent set? Use it to prove that

$$\alpha(G) \geq \frac{|V|^2}{4|E|}. \tag{2}$$

Bonus: Imagine an infinite graph of vertices and edges arranged in a square 2D grid. What's the smallest fraction of vertices needed to cover in order to monitor every vertex in the graph? What if it were a regular triangular grid?

Problem 3: If I Could Turn Back Time

A time eddy from Max's timejump hits unexpectedly, sending BagelBot a random number of minutes nto the past. Time then proceeds normally, until BagelBot returns to the moment the time eddy hit - and with probability p is sent back another random number of minutes.

1) If each timejump is uniformly random, between 1 and T, what's the expected amount of time it will take BagelBot to catch up to the present? Does BagelBot make it back? (How does it depend on p and T?)

2) If each timejump is *geometrically* distributed, with parameter q, what's the expected amount of time it will take BagelBot to catch up to the present? Does BagelBot make it back? (How does it depend on p and q?)

1) If each timejump is uniformly random, between 1 and T, what's the expected amount of time it will take BagelBot to catch up to the present? Does BagelBot make it back? (How does it depend on p and T?)