1_sim_sim.py

1. For all bots which are currently at a node & idle

Update:

- true idleness of all nodes in graph (true) = +Δt
- expected idleness of all edges in the graph (expect) = +Δt

Calculate:

- Value function all edges where bots are present (Q) - = $\frac{(expect - true)}{expect}$

- Softmax of Value function = value_exp = $\frac{e^{Q_i}}{\sum_{j=i}^{k} e^{Q_j}}$ (summation over all edges)

Set:

- True idleness of nodes where bots are present = 0
- Expected idleness of nodes wrt the corresponding bot = 0

NOTE: So effectively, we are calculating expected idleness perceived by the bot as the time elapsed since this bot last visited the node.
This is the **AGENT model** where the bot is calculating expected idleness by seeing the time elapsed since its last visit. Hence there is **no notion of memory** (estimating the expected idleness based on previous visits)

2. For a bot deciding the next node to visit

Set:

- True idleness of the node where the bot is present = 0
- Expected idleness of the node wrt the corresponding bot = 0

Decision Making: here, we chose ε=0.1

- With $(1 - \varepsilon)$probability, check all neighbours and visit the one with highest value of = $[expected\ idleness]\ x\ [value\_exp]$

- With ε probability, go to a random node

----------END----------

DOUBTS:

- Here, there is no need of expected idleness to depend on edge right? It is just a function of time, bot and current node.
- Is it correct to initialize the value function to zero?

MAJOR DRAWBACK:

- The decision making of next node is[$expected\ idleness$] x [$value\_exp$]
  So when expect=true; implies no other bot has visited this node yet, this bot also has no incentive to visit this node and hence, true idleness can go to very high values.