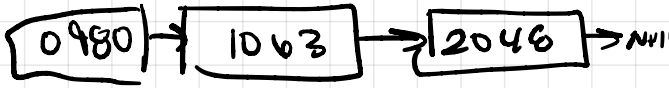
HomeWork 6

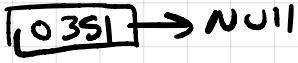
Jakob Symchych
nrz473
4/23/23

1) P vs. NP - Resale Company

∴ Think of databases, as a 2d-Array

// - n people in fig. 1
// - m items in fig. 2
// - e/a customers "Buy" & "Sell" are stored in L of len. at most m

=> ex: WTB = 

WTS = 

1.1) Process Transaction 1

// search through all customers WTS LL for specific item

=> process Transaction 1 ∈ P // $O(n^2)$ to search fig. 1 & $O(m)$ to search LL

• Pseudo code

=> iterate through Fig 1 data base (n time) searching e/a n for a unique item in its LL which has a length $\leq m \in O(n^2 \cdot m)$ runtime

1.2) Process Transaction II

=> Process Transaction II $\in$ P

// Sort all customers in Ascending Order, than Traverse this sorted Array starting at n^{th} element checking if that element is selling the specific item

- **Pseudo code**

=> Create empty 2d Array (A) which is filled w/ fig 1. data, this data is sorted based on customer rating. Traverse A starting at the n^{th} index. We then have to traverse the length m LL of e/a customer until specific item is found

=> Creating sorted 2d Array $\in O(r \cdot c)$ (r = rows, c = columns)

traversing A $\in O(n^2)$

traversing LL $\in O(m)$

$= O(r \cdot c \cdot n^2 \cdot m)$ - still polytime

1.3) Shopping Buddy 1

=> Shopping Buddy 1 $\in$ NP

=> This problem seems similar to the Clique problem. If we let customers be vertices of a graph, and their edges are made up of similar items they wish to buy, we could create a Clique from this created graph which is a known NP-hard problem

1.4) Shopping Buddy II

=> Shopping Buddy II $\in$ NP

=> Same reason shopping buddy I $\in$ NP applies to shopping buddy II.

=> If we let customers be vertices of a graph, and their edges are made up of similar items they wish to buy, we could create a Clique from this created graph which is a known NP-hard problem

1.5) Imposter Names

=> Imposter Name $\in P$

- **Pseudo Code**

=> Iterate through pairs of usernames, checking if they share subsequence of length K which can be done in polynomial time.

1.6) Equal Trade

(seems like subset sum)

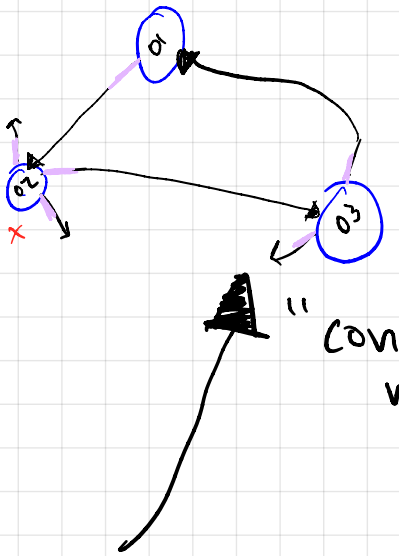
1.7) Gift Exchange

(looks like ham-cycle?)

2.) P vs. NP - Train Company ∴

// direct-Route: 01 → 02 is a direct-route

// Indirect - Route: 01 → 02 → 03 is Indirect as we go to 02 before 03



"connected network"

// Network is connected if every active train station is reachable from every other train station

- n stations
  - M departures per station
- K is parameter user selects