| Tongert common prefix   |
|---|
|   |
| Vo find the longest common prefix among a list of strings, like's   |
| o studette longest common prefix among e list of strings, less's  |
|   |
| Thategy: Vestecal Teaning (character by character):   |
| this is one of the most intentive and officient methods   |
| 17/ 1, Pil # 15+ +-   |
| I sala il 1 viciz ma first suing es a seference.  |
| 2.191 Each Characles, position I in the reference sing:   |
| 1 Idea : [1 Pick the first string as a seference.  2 For each character position i in the reference string:  Compare it with the character of position i in all other  strings  |
| shings. If any string in too shirst, or a character doesn't match, stop 3. Petersu the profess built so fas.  |
| 2 Poten Ha roleic built of la   |
| o, rein me propor Dan Jas.  |
| TExample / Far ["Slover" law" " light"]   |
| · consers 1 set index oin all - 1   |
| · compare / L'at moles 1 -> V   |
| compare o'est midex 2- "flight has i' - x stop  |
| Example: For ["flower" flow" "flight"]  · compare for index oin all -> \  · compare \[ \cap \text{at index } \]  · compare \[ \cap \text{at index } \]  · compare \[ \cap \text{at index } \]  · flight "has i' -> \text{Itop}  Reterm "fl" |
|   |
| Jedge Eeses / emply ingret - return   |
| enly one string - return it<br>- one string is emple - return   |
| - gue string is exister - setuen  |
| - MI common prefere at all - return "c  |
|   |
| Pseudocode Outline:   |
| if star is empty:   |
| Network to C.7  |
| prefix z strs Co7   |
| for each string in stes [1:]:   |
| Levile current string doen to start with prefix:  |
| remove last char from prefix  |

| if prefix lecomes empty:   |  |
|--|--|
| retury 44  |  |
| if prefix lecomes empty:  return prefix  |  |
|  |  |
| This method govereds full character-bu-character comparison and use the  |  |
| This method avoids full character-by-character compasison and uses the "start with "logic efficiently.   |  |
| in the state of th |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |