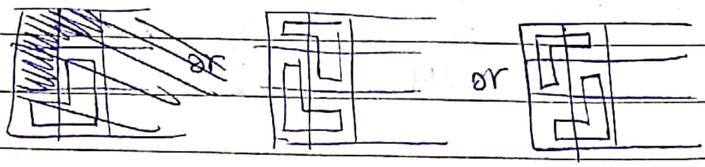
1182

DATE □□ □□ □□□□

(A) Starting from the left there ~~can~~ are only ~~be~~ 2 ways to fill the shapes



both of these reduce the problem to (n-2)

⇒ if n is odd ⇒ not possible

otherwise $2^{n/2}$ is the answer

Time complexity O(1)

(B) *Firstly, all *'s in a row or column must be consecutive

let r(i) : count of *'s in row i

c(i) : count of *'s in column i

let T be the total number of *s

~~for i=1 to H~~ if there exists any cell (i,j) such that

~~for j=1~~ it contains a '*' and r(i) + c(j) - 1 = T

then we return true, otherwise false

Time complexity O(hw)

(C) First group the words according to number of vowels

(let them be $G_1, G_2 \ldots G_K$)

for each group G_i, we can make $f_i = \left\lfloor \frac{|G_i|}{2} \right\rfloor$ pairs that can be used as first words in both lines

~~This does not maximize~~ Note that if a pair can be used as second word pair

Then it can also be used as first word pair.

⇒ We divide $G_i \rightarrow G'_{i_1}, G'_{i_2} \ldots$ (grouped ~~according to~~ by same last vowel)

$\hookrightarrow G''_i$ (rest of the words)

→ Note that we can still make f_i pairs, but we have maximized second word pairs (by making second word pairs first)

let F → first word pairs, S → second word pairs

if |S| > |F| + 1, then we ~~take~~ remove some pairs from S and put them in F until this condⁿ is not satisfied

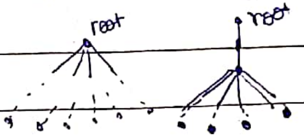
Time complexity : O(nlogn)

DATE

(D) Let's think ~~of~~ about the diameter of this tree, ~~if the diameter is even~~

→ if the diameter is ~~even~~ odd, then ~~[illegible]~~ one of the endpoints ~~may~~ must be root

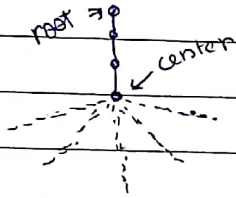
Why? Because all leaves are equidistant from the root (⇒ even diameter if any two leaves are endpoints)
which are not root



⇒ if the diameter is even, then ~~either~~ two possibilities

① center is the root (~~non~~ root is not a leaf)

in this case root is a leaf

② 

(Why linear chain from center?)

↳ Because if there is some branch in this chain, it has to be identical to other children of that subtree

⇒ this would change the center

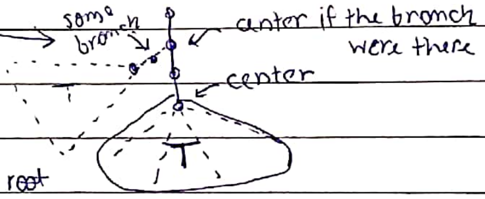
* Now, it may be the case that

→ all branches are linear chains

→ all may be of same depth

→ pick any, return leaf as root

→ one is different ⇒ return its leaf as root



→ otherwise only 1 branch must be a linear chain, and ~~root~~ we return its leaf as root

Note: If none of the cases are met, ~~there~~ root doesn't exist.

Note 2: Before returning any node as root, we check it first by DFS & degree check (verification)

Time complexity: O(n)