Algorithm:

Q.S. $R = \{s\}$

Perform BFS

check if any bases in layer are flag base

Recursively check if base is in next layer for each node

Add "shortest" (least expensive) path to R

Return R

Analysis: BFS takes $O(m+n)$ as established in class

Recursively BFS, n $O(m+n)$ has n times

for each node (n nodes) we traverse its edges (m edges)

$O(m+n)$

Proof: By recursively BFS-ing we can check for least expensive path at each node and each time we will treat the next node as the new root. If we store each path in a DS. we will be able to compare paths and backtrack our way up the stack trace selecting the least expensive path from each call.

Combining each least expensive path will lead to us creating the ~~the~~ least expensive path from home base to the enemy's flag base