1. computePath() will be used in nextPosition()
2. Fairy parameters for:
    a. Start point - this should be the current Point that the fairy is in (fairy.position)
    b. End point - this should be the position of the entity that the fairy is trying to get to (target.position), which should most likely be a stump
    c. canPassThrough - We have to check to make sure that the point is within the bounds of the world and that it is also not a position that we are already in or one we have already visited in this path. We also have to check if the block we are trying to move to is an obstacle that we can't pass through, such as the river, or if it is already full.
    d. We could probably use the moveTo() method. Specifically looking at Fairy, we can also see that this method uses the adjacent() method to see if we are next to the target position, which is what we are looking for.
    e. potentialNeighbors: PathingStrategy.CARDINAL_NEIGHBORS
3. One thing that changes for the dude's pathfinding is that the dudes can move through stumps while the fairies can't. Instead, the fairies move to the stump and do their fairy actions to make the stumps regrow into trees.
4. If we don't find any valid spaces to move to, we just set newPos to the position that the entity is currently in. In other words, if we can't find anything, we just stay in the same place. In computePath, we would do nothing if there are no valid spaces to move to. We could have computePath return a boolean, which would be returned as false in this case.