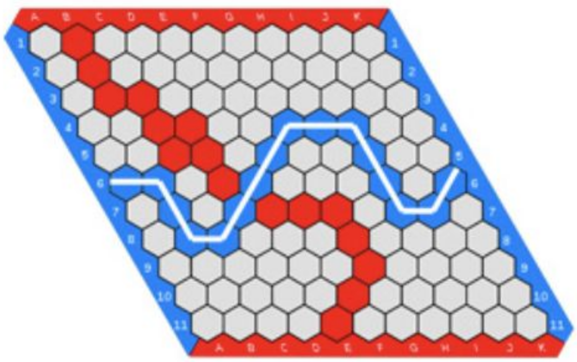
- Problem: Hex is a two player abstract strategy board game in which players attempt to connect opposite sides of a hexagonal board. Hex was invented by mathematician and poet Piet Hein in 1942 and independently by John Nash in 1948. (Wikipedia)

  Design an algorithm that checks if the winner exists after every move.



Solution: The board can be seen as a graph in which vertices are hexagons and edges connect hexagons that share a wall. Furthermore, add 4 artificial vertices, two for each player representing the sides and connect them using edges with all hexagons on a given side. After every move call union on the changed vertex and check using find if the two artificial vertices are connected.

- Problem: Suppose we want to make change for n cents, using the least number of coins of denominations 1, 10, and 25 cents. Consider the following greedy strategy: suppose the amount left to change is m; take the largest coin that is no more than m; subtract this coin's value from m, and repeat. Either give a counterexample, to prove that this algorithm can output a non-optimal solution, or prove that this algorithm always outputs an optimal solution.

  We give a counterexample demonstrating that the greedy algorithm does not yield an optimal solution. Consider the problem of making change for 30 cents. The optimal solution is to give three dimes. However, the greedy algorithm first chooses a 25 cents piece, and then is then forced to use 5 pennies, leading to a non-optimal solution.

Exercise 3. Suppose that we have a set $S = \{a_1, \ldots, a_n\}$ of proposed activities. Each activity a_i has a start time s_i and a finish time f_i. We can only run one activity at a time. Your job is to find a maximal set of compatible activities. Which of the following greedy algorithms is correct?

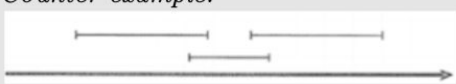
(a) Sort all the activities by their duration and greedily picking the shortest activity that does not conflict with any of the already chosen activities.

(b) Pick the activity that conflicts with the fewer number of remaining activities. Remove the activities that the chosen activity conflicts with. Break ties arbitrarily.

(c) Sort all the activities by their end time and greedily pick the activity with the earliest end time that does not conflict with any of the already chosen activities.

Solution

(a) *Incorrect.* It's not always best to choose the shortest length activities
Counter-example:



(b) *Incorrect.* This one is tricky, but it's not always best to choose the activity with the least amount of conflicts.
Counter-example:

(c) *Correct.*

Proof: Let g_1 be the first activity chosen by the greedy solution and let t_1 be the first activity by the optimal solution. We know that based on our selection technique, g_1 is has the earliest end-time out of all the activities, and thus the end time of t_1 must be equal to or after that of g_1. A similar argument can be made for t_2 and g_2. We know that t_2 does not overlap with g_1 because g_1 ended before t_1 ended. Therefore t_2 must end at the same time or later than g_2. This argument can be made inductively to show that the end time of g_i is always earlier than the end time of t_i.

Why does this imply that the greedy algorithm is optimal? Imagine a situation where we have one more activity in the optimal solution than in greedy. Let t_ℓ be the last event chosen by the optimal solution, but suppose that greedy only goes up to $g_{\ell-1}$. Well we know by the informal induction we did that the end time of $g_{\ell-1}$ is no later than the end time of $t_{\ell-1}$. This implies that t_ℓ does not conflict with $g_{\ell-1}$. So why did the greedy algorithm not choose t_ℓ after choosing $g_{\ell-1}$? Contradiction.