HINTS ON PROBLEM 2

Useful functions to have

1) Compute-worst-drive-time / given a choice of distribution centers

- find relevant drive times
- find closest drive time for each hospital given the current choice of distribution centers
- find the worst drive time out of the previous ones

UPDATE STEP

2) Find-best-single-hospital-replacement

```
for all distribution centers / in your current selection
    for all hospitals — in the hospital list
        if hospital is not among the
        current choice of distr. centers
            → Replace d.c with hospital
            → compute_worst-drive-time
            → If new worst drive time
              has "improved", you have
              found a better candidate list.
```

may need to break ties

Out of all the "better candidates" keep the one that improves worst drive time the most keep calling the UPDATE STEP until you see no improvement in the worst drive time

EXAMPLE

drive time dataframe

	A	B	C	D	E
A	0	1	3	4	5
B	1	0	7	1	8
C	3	7	0	15	2
D	4	1	15	0	7
E	5	8	2	7	0

5 hospitals

→ choose 2

distribution centers

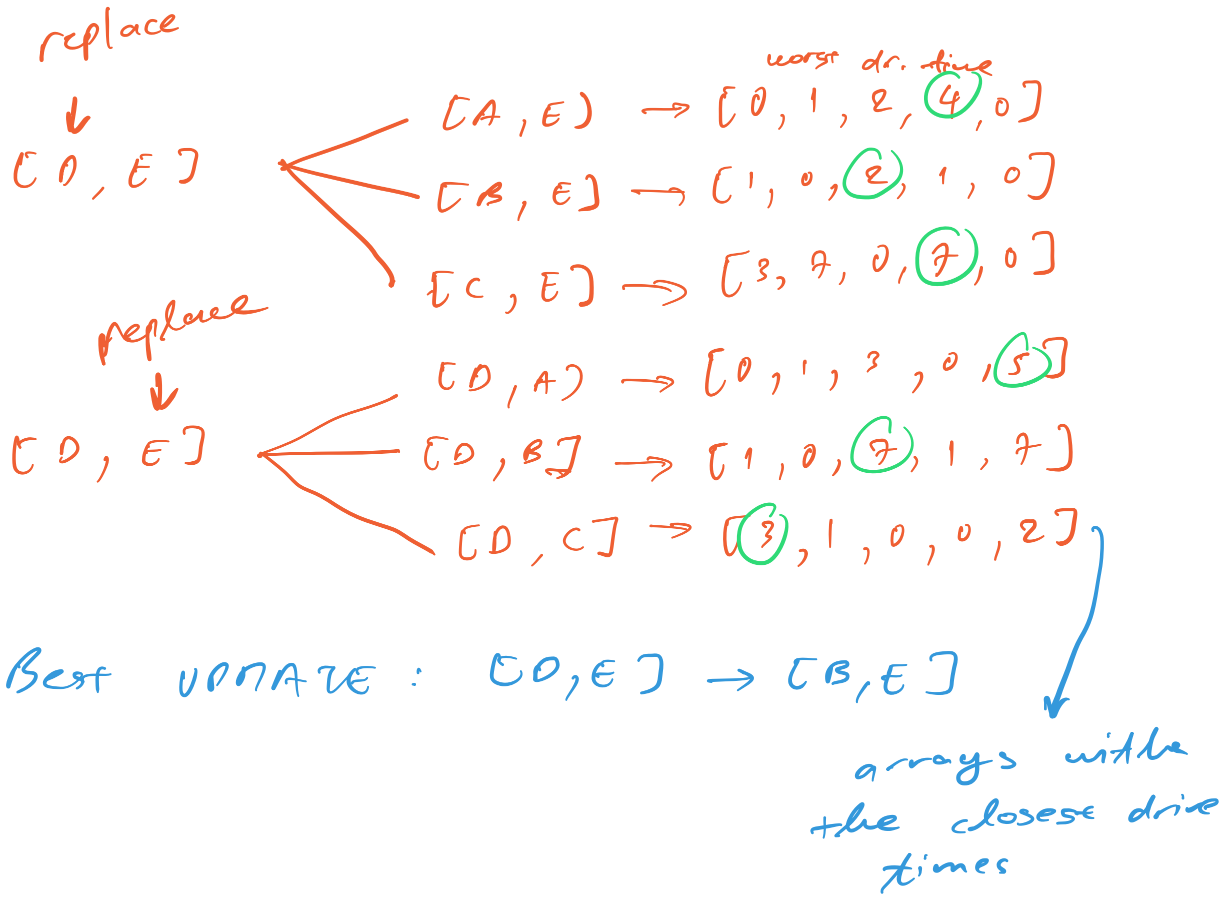
Initial feasible solution [D, E]

Run one iteration of

find-best-single-hospital-replacement

Can only replace one center at a time

```
for all distribution centers [D, E]
    for all hospitals [A, B, C, D, E]
        if hospital is not among the    [A, B, C]
        current choice of distr. centers
            → Replace d-c with hospital
            → compute_worst_drive_time
            → If new worst drive time
              has improved, you have
              found a better candidate list.
```



Best UPDATE : [D, E] → [B, E]