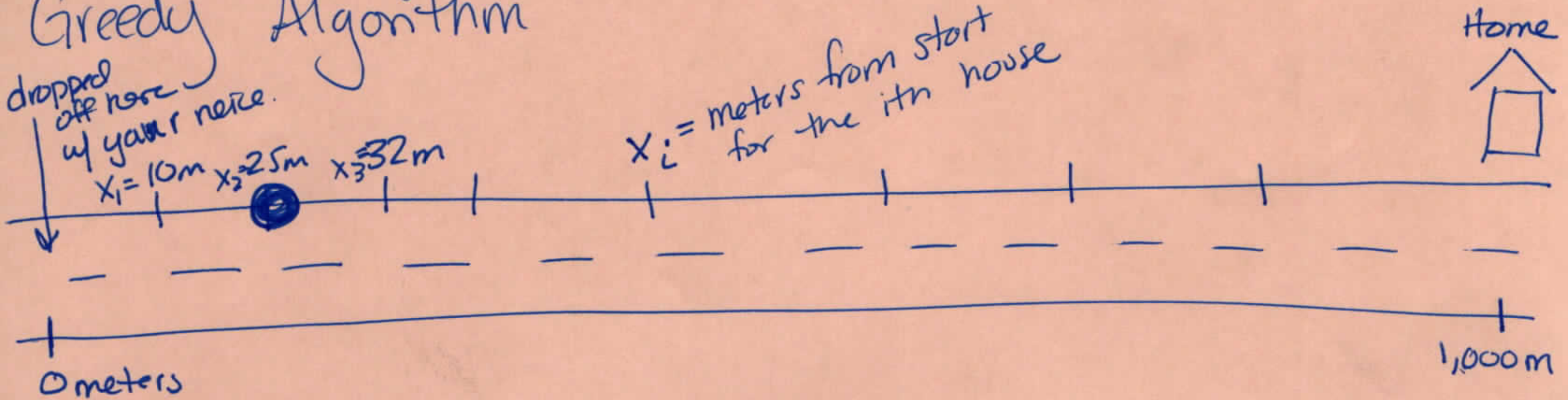
8 Oct 2021

Misc Next week

1- Thurs AM IEEE vis Q&A
2. Postdoc visiting (maybe)
3. pregame broadcast on Fri 15th 7:15

Greedy Algorithm



It's Halloween:

→ you need to take neice trick or treating (she'll only walk 30 meters between stops, at most)

→ you want to get to party ASAP

→ need to get home to leave neice w/ grand parents + join your sister at a party.

Greedy strategy

→ from current pos, look ahead 30m & check closest house not past 30 m.

→ repeat until we get home.

Questions

① Is this algorithm correct? Does this get me home w/out going more than 30 m between consec. homes?

② Is this algorithm optimal? Does this find the sol'n w/ the fewest # of stops. ①

Algorithm

input: $X = \{x_i\}_{i=1}^{k}$, ordered list of distances to houses
Note that home is at x_k.

output: $Y = [y_1, \dots, y_{k'}]$ is an ordered subsequence of X that optimally solves the problem.

Must Assume

① $x_1 \le 30$
② $x_i - x_{i-1} \le 30$

```
pre append x_0 = 0 to X ; init Y = []
curindex = 0
next index = 1
while nextindex ≤ k
    while x_{nextindex+1} - x_{curindex} ≤ 30
        ++ next index
    endwhile
    ~~add~~ add x_{nextindex} to Y
    cur index = nextindex
    next index = curindex + 1
endwhile
return Y
```

Post condition: Y is a sequence of distances to houses such that all adjacent houses are at most 30 m apart, Y includes x_k, and $y_1 \le 30$.

What is the loop invariant?
L_i is the statement: Y is a subsequence of X that successfully gets from start to $x_{curindex}$,
nextindex = curindex + 1
→ what does this mean?
→ Y is a subseq. of X
and 1st elt. of $Y \le 30$ m from start
and $|y_i - y_{i+1}| \le 30$ for all $i < |Y|$

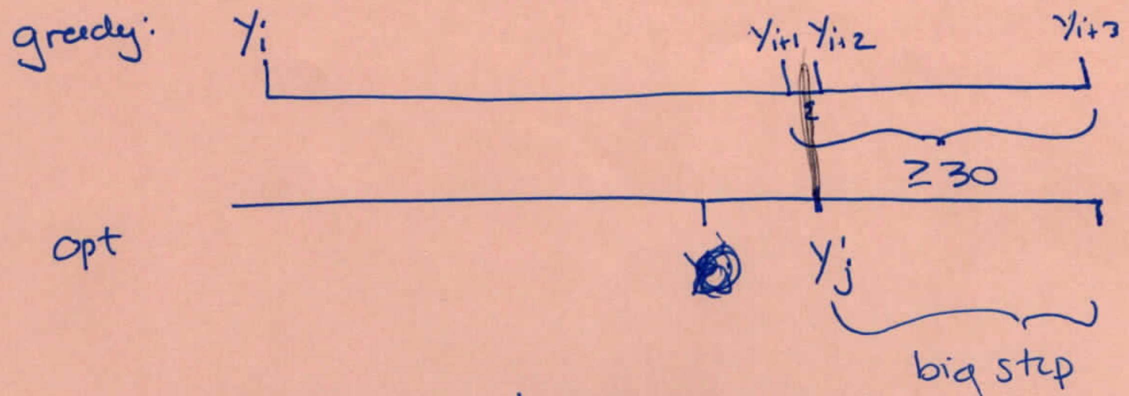
Y_G = the sol'n from our greedy algorithm

Y_O = an arbitrary optimal sol'n.

$\boxed{\text{WTS: } |Y_G| \le |Y_O|}$ (as we know, by def'n of optimal $|Y_O| \le |Y|$ for any sol'n Y + hence $|Y_O| \le |Y_G|$)

How?

- You could never take a bigger step.



potential prob: the greedy choice locked us into another choice. Need to show that things can't go south!

- the greedy choice "stays ahead" of any other solution.

$$Y_G = [y_1^G, y_2^G, \ldots\ldots, y_n^G]$$

$$Y_O = [y_1^O, y_2^O, \ldots\ldots, y_m^O]$$

show: $\boxed{y_i^G \ge y_i^O}$

Proof that Y_G "stays ahead" of Y_O:

Base case: Covered the 1st time we encounter the while loop. We select y_1^G as the largest $x_i \le 30$, since currIndex = 0

$\Rightarrow$ y_1^O must be $\le y_1^G$

(if larger, would have been bigger than 30m).

Ind. Assump: ~~Assume y.~~ Let $i \ge 1$. Assume $y_i^O \le y_i^G$.

Ind. Step. NTS: $y_{i+1}^O \le y_{i+1}^G$

Do this by walking through the algorithm