Design and Analysis of Algorithms

02-01
Greedy Algorithms

Main Ingredients of Greedy Algorithms

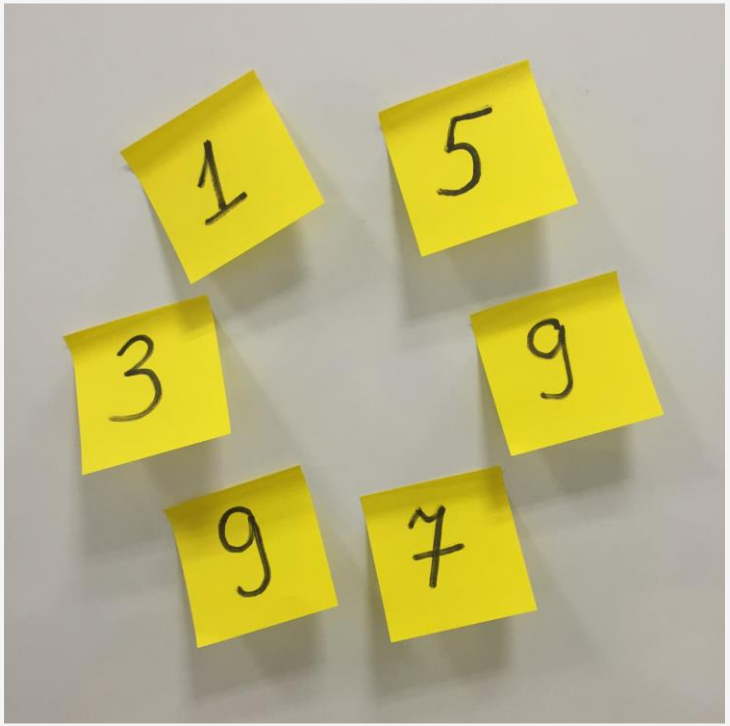
Imran Ihsan
Assistant Professor, Department of Computer Science
Air University, Islamabad, Pakistan
www.imranihsan.com

Largest Number

Learning Objective:

Produce a greedy algorithm yourself

Job Interview



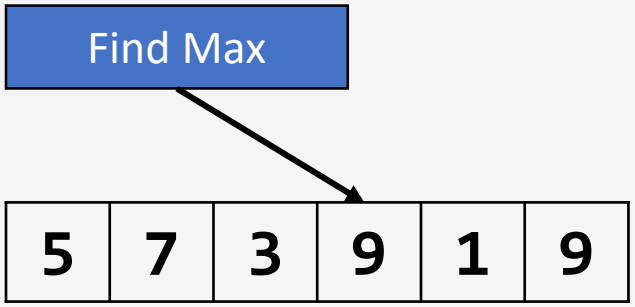
Largest Number

What is the largest number that consists of digits 3, 9, 5, 9, 7, 1? Use all the digits.

Greedy Strategy

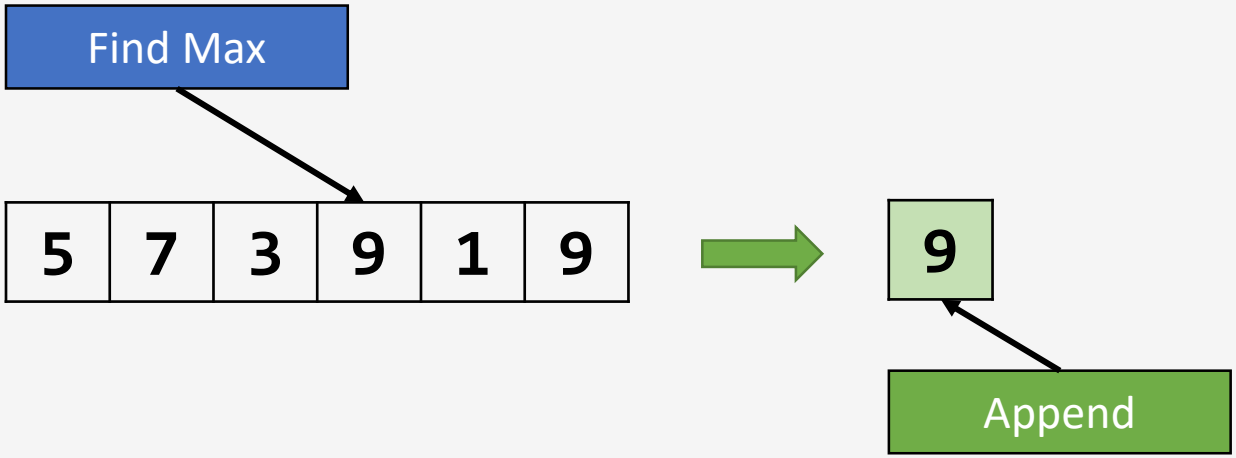
5	7	3	9	1	9

Greedy Strategy



Find max digit

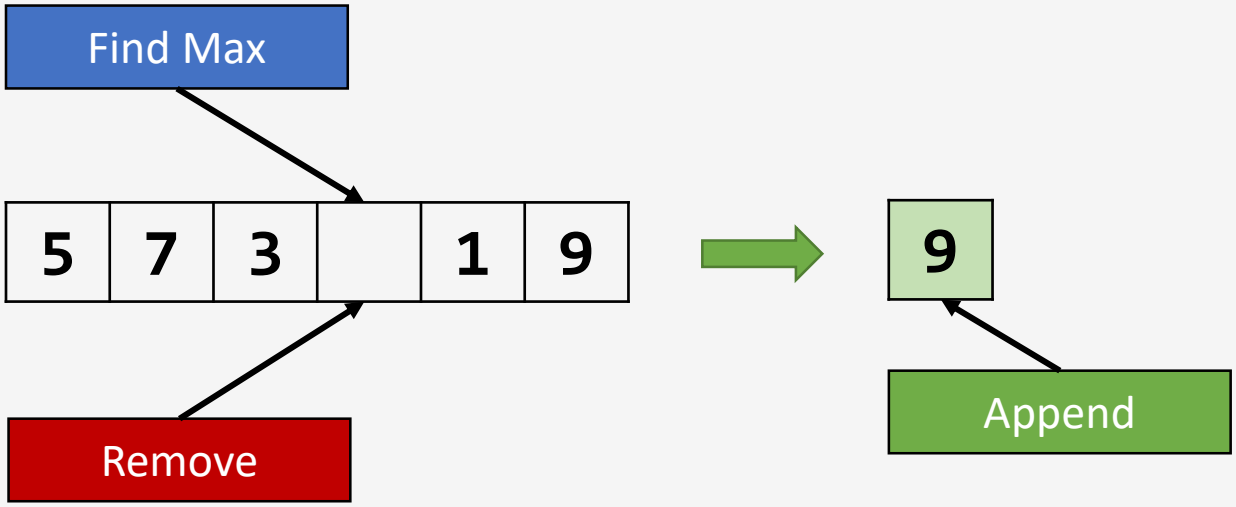
Greedy Strategy



Find max digit

Append it to the number

Greedy Strategy

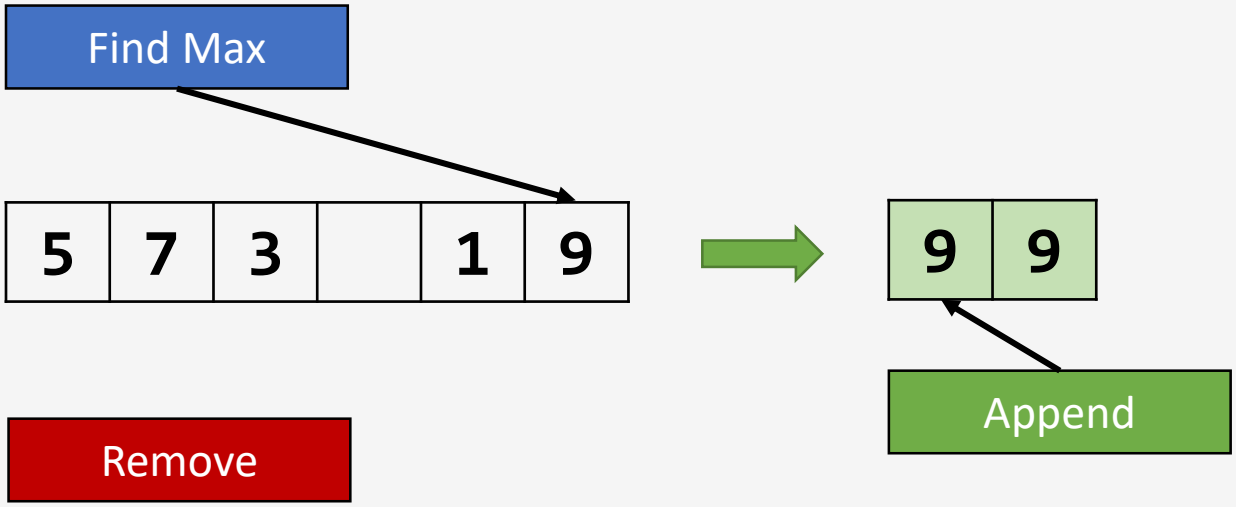


Find max digit

Append it to the number

Remove it from the list of digits

Greedy Strategy



Find max digit

Append it to the number

Remove it from the list of digits

Repeat while there are digits in the list

Greedy Strategy

Find max digit

Append it to the number

Remove it from the list of digits

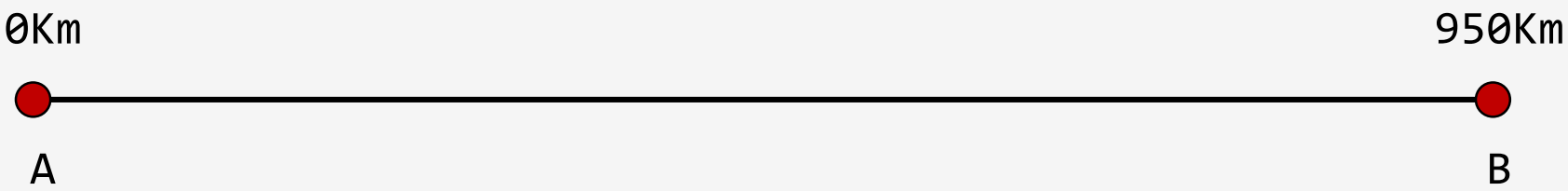
Repeat while there are digits in the list

Car Fueling

Distance with full tank = 400km

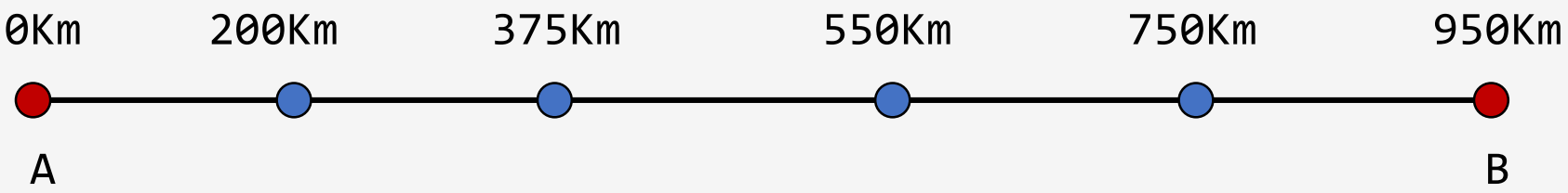
Car Fueling

Distance with full tank = 400km



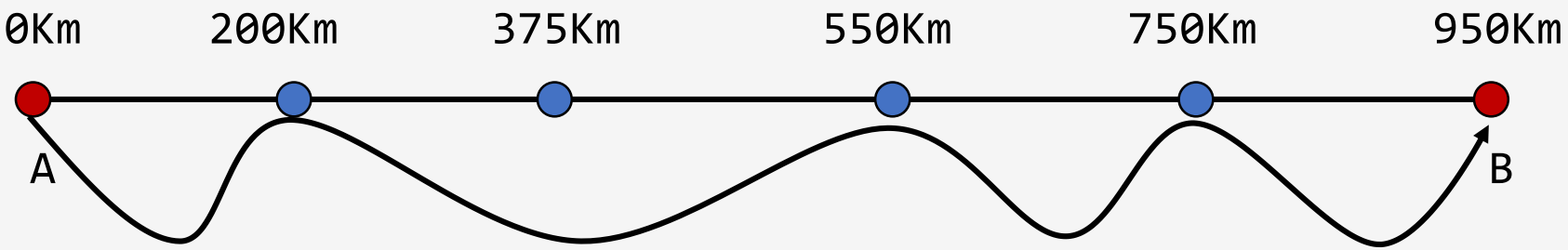
Car Fueling

Distance with full tank = 400km



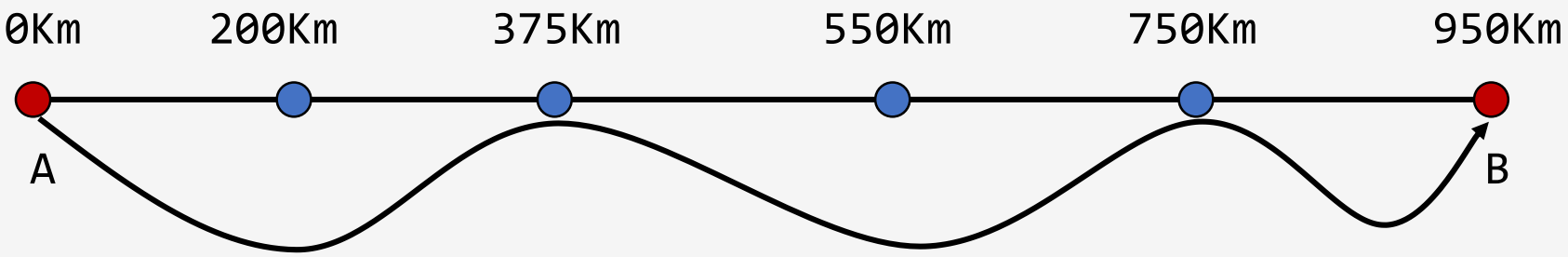
Car Fueling

Distance with full tank = 400km



Car Fueling

Distance with full tank = 400km



Minimum number of refills = 2

Car Fueling

Input:

A car which can travel at most L kilometers with full tank,
a source point A, a destination point B and
n gas stations at distances $x_1 \leq x_2 \leq x_3 \leq \cdots \leq x_n$ in kilometers
from A along the path from A to B.

Output:

The minimum number of refills to get from A to B, besides refill at A.

Greedy Strategy

Make some greedy choice

Reduce to a smaller problem

Iterate

Greedy Choice

Refill at the closest gas station

Refill at the farthest reachable gas station

Go until there is no fuel

Greedy Algorithm

Start at A

Refill at the farthest reachable gas station G

Make G the new A

Get from new A to B with minimum number of refills

Subproblem: Definition

Subproblem is a similar problem of smaller size.

Subproblem: Example

LargestNumber(3, 9, 5, 9, 7, 1) = “9” + LargestNumber(3, 5, 9, 7, 1)

Min number of refills from A to B = first refill at G + min number of refills from G to B

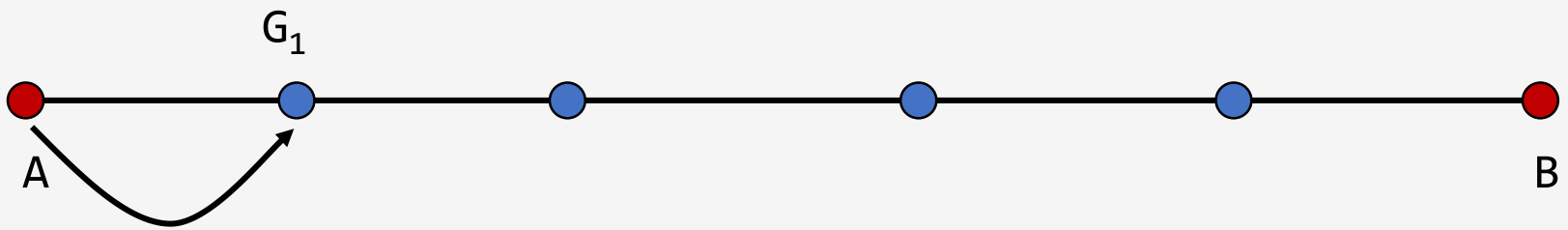
Safe Move: Definition

A greedy choice is called safe move if there is an optimal solution consistent with this first move.

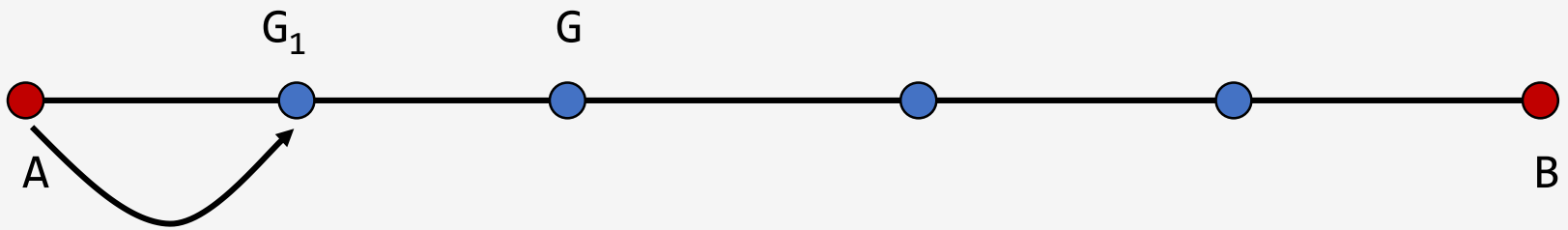
To refill at the farthest reachable gas station is a safe move.

Proof

Proof

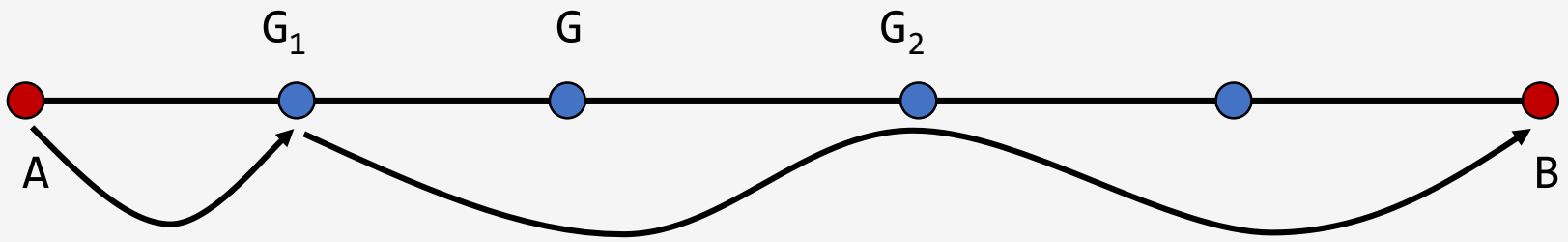


Proof



First case: G is closer than G2

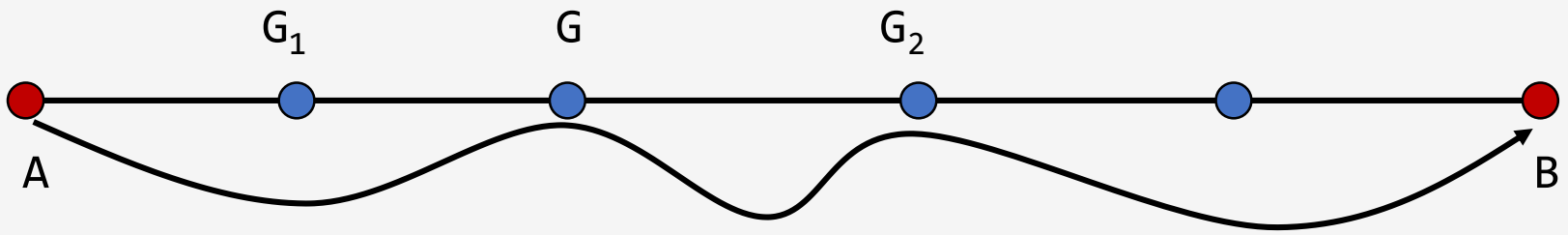
Proof



First case: G is closer than G2

Refill at G instead of G1

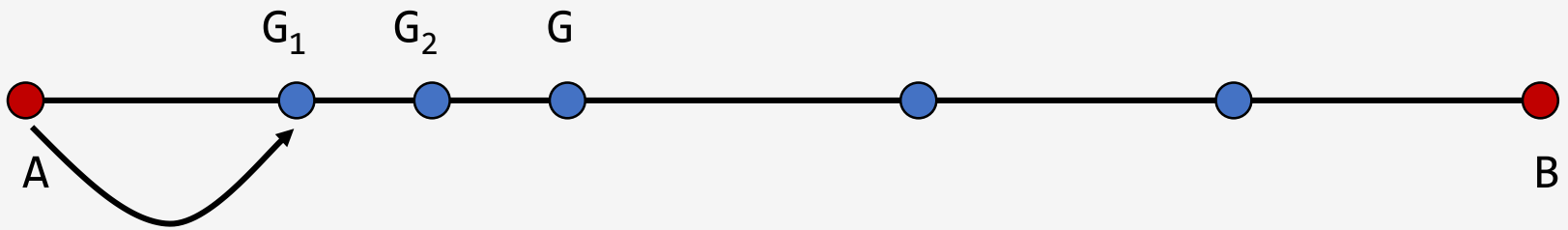
Proof



First case: G is closer than G2

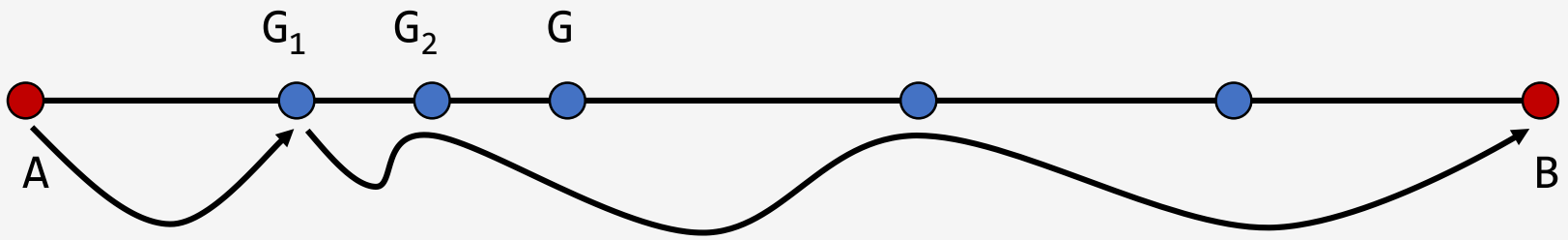
Refill at G instead of G1

Proof



Second case: G2 is closer than G

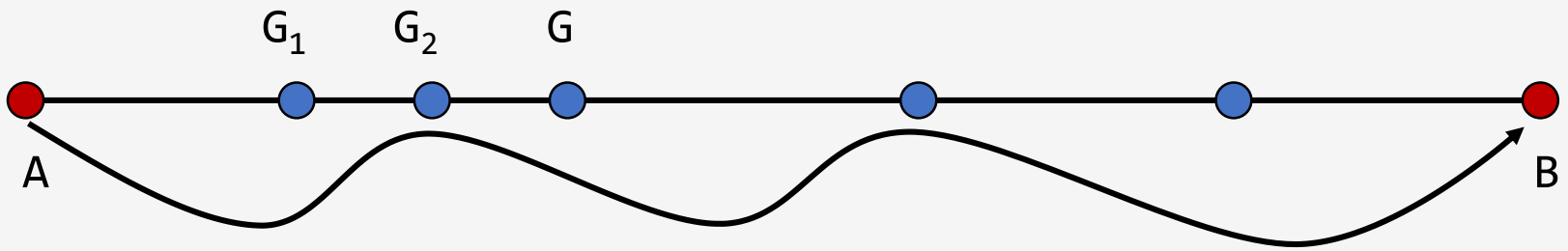
Proof



Second case: G2 is closer than G

Avoid refill at G1

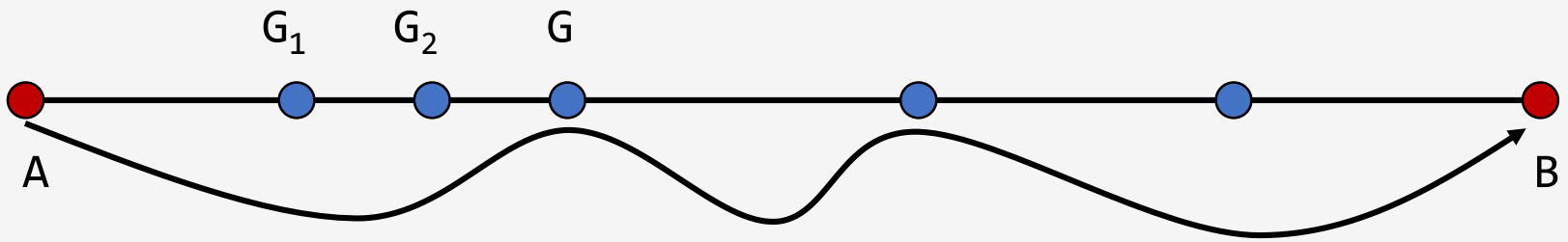
Proof



Second case: G2 is closer than G

Avoid refill at G1

Proof



Second case: G2 is closer than G

Avoid refill at G1

Proof

Route R with the minimum number of refills

G1 → position of first refill in R

G2 → next stop in R (refill or B)

G → farthest refill reachable from A

If G is closer than G2, refill at G instead of G1

Otherwise, avoid refill at G1

Implementation and Analysis

MinRefills

$$A = x_0 \le x_1 \le x_2 \le \cdots \le x_n \le x_{n+1} = B$$

```
MinRefills(x, n, L)
        numRefills ← 0, currentRefill ← 0
        while currentRefill ≤ n:
            lastRefill ← currentRefill
            while (currentRefill ≤ n and
                x[currentRefill + 1] – x[lastRefill ] ≤ L):
            currentRefill ← currentRefill + 1
            if currentRefill == lastRefill:
                return IMPOSSIBLE
            if currentRefill ≤ n:
                numRefills ← numRefills + 1
        return numRefills
```

MinRefills

Lemma

The running time of MinRefills(x, n, L) is O(n).

Proof

currentRefill changes from 0 to n + 1, one-by-one

numRefills changes from 0 to at most n, one-by-one

Thus, O(n) iterations

Main Ingredients

Reduction to Subproblem

Make a first move

Then solve a problem of the same kind

Smaller: fewer digits, fewer fuel stations

This is called a “subproblem”

Safe Move

A move is called safe if there is an optimal solution consistent with this first move

Not all first moves are safe

Often greedy moves are not safe

General Strategy

Make a greedy choice

Prove that it is a safe move

Reduce to a subproblem

Solve the subproblem