ECS 122A: Algorithm Design and Analysis

Week 5 Discussion

Apr 28, 2021

Roadtrip Refueling Problem

Problem Statement: Suppose you plan a long-haul roadtrip. The tank capacity of your car is limited, so you look at the route ahead of time and know the locations of the gas station along the way. The question is how to make as few refueling stops as possible (assuming tank is always fueled full-to-full)?

Roadtrip Refueling Problem

Problem Statement: Suppose you plan a long-haul roadtrip. The tank capacity of your car is limited, so you look at the route ahead of time and know the locations of the gas station along the way. The question is how to make as few refueling stops as possible (assuming tank is always fueled full-to-full)?

Input:

1. $d[1 ... n]$: refueling stops such that $d_1 < d_2 < \cdots < d_n$ where $d[1]$ is the start and $d[n]$ is the destination
2. c: tank capacity

Output: the minimum number of refuels

Roadtrip Refueling Problem: Greedy Strategy

Recall greedy algorithm always makes the choice that looks best at the moment, without regard for future consequence.

What is the most intuitive greedy strategy?

Roadtrip Refueling Problem: Greedy Strategy

Recall greedy algorithm always makes the choice that looks best at the moment, without regard for future consequence.

What is the most intuitive greedy strategy?

Go as far as possible within fuel capacity!

Example

- $d = [0, 220, 400, 550, 760, 950, 1100, 1180, 1280, 1350, 1500]$
- $c = 300$

We would stop at $220, 400, 550, 760, 950, 1180, 1350$ miles.

Roadtrip Refueling Problem: Pseudocode

```
REFUELING(d, c)
S = ∅
curr = 0
while curr! = d[n]
    d[s] = greatest number such that d[s] ≤ curr + c
    if d[s] = curr
        return "no solution"
    else
        S = S ∪ {d[s]}
        curr = d[s]
return S
```

Roadtrip Refueling Problem: Pseudocode

REFUELING(d, c)

```
S = ∅
curr = 0
while curr != d[n]
    d[s] = greatest number such that d[s] ≤ curr + c
    if d[s] = curr
        return "no solution"
    else
        S = S ∪ {d[s]}
        curr = d[s]
return S
```

Time complexity: $O(n)$

Roadtrip Refueling Problem: Greedy Choice Property [1]

1. Let the set of optimal solution be $O = \{o_1, o_2, \cdots, o_i\}$ where $o_1 < o_2 < \cdots < o_i$.
2. We have another valid set chosen by the greedy strategy $G = \{g_1, g_2, \cdots, g_j\}$ such that $g_1 < g_2 < \cdots < g_j$. j might be greater than i.
3. We assume that g_k $(1 \leq k \leq j)$ is the first choice in G that differs from the stop in O. Thus, we have a new set $O' = O - \{o_k\} + \{g_k\}$.
4. $g_k > o_k$ *("greedy" choice)*
   $o_k + c \geq o_{k+1}$ *(reachable from o_k to o_{k+1})*
   $\implies g_k + c > o_{k+1}$ *(transitivity of inequalities)*
   $\implies$ All the stops in O' are valid
5. Lastly, since $|O'| = |O| - 1 + 1 = |O|$, O' is also optimal.

[1] The proof of optimal substructure property is left to the readers.