COMP3711(H): Design and Analysis of Algorithms

Tutorial 6: Greedy

GY1

Greedy Interval Covering

Problem

A *unit-length closed interval* on the real line is an interval $[x, x+1]$.
Describe an O(n) algorithm that, given input set

$$X = \{x_1, x_2, \dots, x_n\}$$

determines the smallest set of unit-length closed intervals that contains all of the given points.

Argue that your algorithm is correct. You should assume that

$$x_1 < x_2 < \cdots < x_n$$

As an example the points above are given on a line and you are given the length of a 1-unit interval.

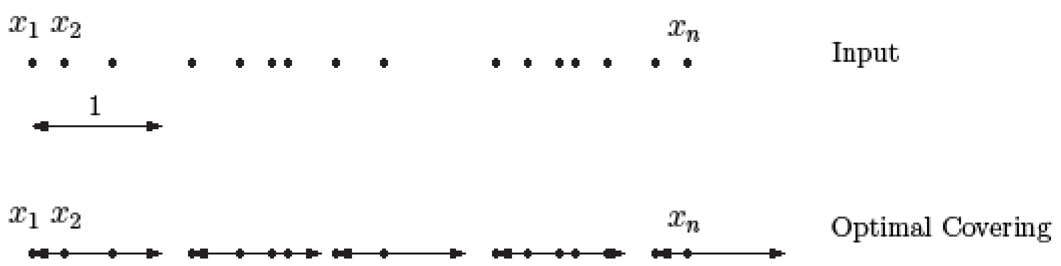
Show how to place a minimum number of such intervals to cover the points.

Solution

Keep the points in an array. Walk through the array as follows.

1. Set $x = x_1$
2. Walk through the points in increasing order
   until finding the first j such that $x_j > x + 1$
3. Output
4. If there was no such j in (2) then stop. Otherwise, set $x = x_j$
5. Go to Step (2)

Since each point is seen only once, this is an O(n) algorithm.



The diagram above is the solution for the points on the previous page

Solution

Keep the points in an array. Walk through the array as follows.

1. Set $x = x_1$
2. Walk through the points in increasing order
   until finding the first j such that $x_j > x + 1$
3. Output
4. If there was no such j in (2) then stop. Otherwise, set $x = x_j$
5. Go to Step (2)

We must now prove correctness.

Let *Greedy(i,k)* be the algorithm run on the Array $[i \dots k]$.
Note that this can be rewritten

1. Output $[x_i, x_i + 1]$
2. Find min j such that $x_j > x_i + 1$
3. If such a j does not exist, stop, else return *Greedy(j,k)*

We will prove correctness by **induction on number of points $|X|$**.

Solution

1. Output $[x_i, x_i + 1]$
2. Find min j such that $x_j > x_i + 1$
3. If such a j does not exist, stop, else return *Greedy(j,k)*

We will prove correctness by induction on number of points $|X|$.

(i) If $|X| = 1$ (only one point) then algorithm is obviously correct.

Otherwise, suppose $|X| = n$ and that we know (induction hypotheses) that the algorithm is correct for **all** problems with size $|X| < n$.

(ii) $|X| = n$ and we know that the algorithm is correct for **all** problems with $|X| < n$.

Solution

1. Output $[x_i, x_i + 1]$
2. Find min j such that $x_j > x_i + 1$
3. If such a j does not exist, stop, else return *Greedy(j,k)*

Proving correctness by induction on number of points $|X|$.

(ii) $|X| = n$ and we know that the algorithm is correct for **all** problems with $|X| < n$.

$O(i,j) = \ minimum\ \#\ \ of\ intervals\ needed\ to\ cover\ \ \{x_i, \ldots, x_j\}$

$G(i,j) = \ \#\ \ of\ intervals\ Greedy\ uses\ to\ cover\ \ \{x_i, \ldots, x_j\}$

Assume $[x_1, x_1 + 1]$ does not cover all of X because, if it did,
Greedy would return that same one interval solution which is optimal.

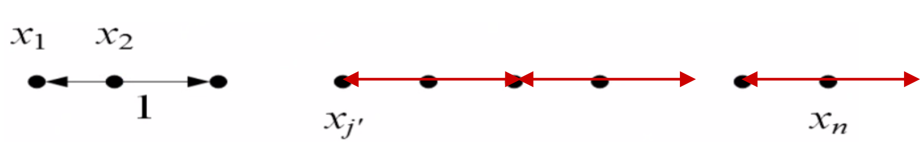
Let j' be the smallest index such that $x_{j'} > x_1 + 1$

Greedy returns $[x_1, x_1 + 1]$ concatenated with the greedy solution for $\{x_{j'}, \ldots, x_n\}$

=> Greedy uses $1 + G(j', n)$ intervals

$[x_1, x_1 + 1]$ Greedy solution for $\{x_{j'}, \ldots, x_n\}$

Example:



Solution

1. Output $[x_i, x_i + 1]$
2. Find min j such that $x_j > x_i + 1$
3. If such a j does not exist, stop, else return *Greedy(j,k)*

Proving correctness by induction on number of points $|X|$.

(ii) $|X| = n$ and we know that the algorithm is correct for **all** problems with $|X| < n$.

$O(i, j) = \ minimum\ \#\ \ of\ intervals\ needed\ to\ cover\ \ \{x_i, \ldots, x_j\}$

$G(i, j) = \ \#\ \ of\ intervals\ Greedy\ uses\ to\ cover\ \ \{x_i, \ldots, x_j\}$

Assume $[x_1, x_1 + 1]$ does not cover all of X because, if it did,
Greedy would return that same one interval solution which is optimal.

Let j' be the smallest index such that $x_{j'} > x_1 + 1$

Greedy returns $[x_1, x_1 + 1]$ concatenated with the greedy solution for $\{x_{j'}, \ldots, x_n\}$

=> Greedy uses $1 + G(j', n)$ intervals

Since $j > 1$, Induction Hypothesis implies that $G(j', n) = O(j', n)$

=> Greedy uses $1 + G(j', n) = 1 + O(j', n)$ intervals

Solution

1. Output $[x_i, x_i + 1]$
2. Find min j such that $x_j > x_i + 1$
3. If such a j does not exist, stop, else return *Greedy(j,k)*

$O(i,j) =$ $minimum\ \#\ of\ intervals\ needed\ to\ cover\ \{x_i, \dots, x_j\}$
$G(i,j) =$ $\#\ of\ intervals\ Greedy\ uses\ to\ cover\ \{x_i, \dots, x_j\}$

=> Greedy uses $1 + O(j', n)$ intervals

j' = smallest index such that $x_{j'} > x_1 + 1$

Will now use this fact to prove $\mathrm{G}(1, n) \le O(1, n)$.
Since, by definition of optimal, $O(1, n) \le G(1, n)$,
this implies that $\mathrm{G}(1, n) = O(1, n)$, i.e., that Greedy is Optimal.

Solution

1. Output $[x_i, x_i + 1]$
2. Find min j such that $x_j > x_i + 1$
3. If such a j does not exist, stop, else return *Greedy(j,k)*

$O(i,j) =$ *minimum # of intervals needed to cover* $\{x_i, \dots, x_j\}$
$G(i,j) =$ *# of intervals Greedy uses to cover* $\{x_i, \dots, x_j\}$

=> Greedy uses $1 + O(j', n)$ intervals

j' = smallest index such that $x_{j'} > x_1 + 1$

Suppose there exists optimal solution OPT different from Greedy.

Let $[x, x + 1]$ be interval in OPT with the **leftmost** starting point.
$x \le x_1$ because otherwise x_1 would not be covered by any interval in OPT.

Example:

Right !

Wrong !

Solution

1. Output $[x_i, x_i + 1]$
2. Find min j such that $x_j > x_i + 1$
3. If such a j does not exist, stop, else return *Greedy(j,k)*

$O(i,j) = \text{ minimum \# of intervals needed to cover } \{x_i, \dots, x_j\}$
$G(i,j) = \text{ \# of intervals Greedy uses to cover } \{x_i, \dots, x_j\}$

=> Greedy uses $1 + O(j', n)$ intervals

j' = smallest index such that $x_{j'} > x_1 + 1$

Suppose there exists optimal solution OPT different from Greedy.

Let $[x, x + 1]$ be interval in OPT with the **leftmost** starting point.
$x \leq x_1$ because otherwise x_1 would not be covered by any interval in OPT.

Let k be the minimum index such that $x_k > x + 1$
After removing $[x, x + 1]$, remaining intervals in OPT must form optimal solution for $\{x_k, \dots, x_n\}$, otherwise we could build better solution using fewer intervals.

=> Optimal uses $1 + O(k, n)$ intervals

Solution

1. Output $[x_i, x_i + 1]$
2. Find min j such that $x_j > x_i + 1$
3. If such a j does not exist, stop, else return *Greedy(j,k)*

$O(i,j) = minimum\ \#\ of\ intervals\ needed\ to\ cover\ \{x_i, \ldots, x_j\}$
$\mathrm{G}(i,j) = minimum\ \#\ of\ intervals\ Greedy\ uses\ to\ cover\ \{x_i, \ldots, x_j\}$

Greedy uses $1 + O(j', n)$ intervals

j' = smallest index such that $x_{j'} > x_1 + 1$

Optimal uses $1 + O(k, n)$ intervals

$[x, x + 1]$ is leftmost interval in OPT
k = smallest index such that $x_k > x + 1$

Because $x \leq x_1$, $k \leq j'$ (Why?)

=> the optimal solution for $\{x_k, \ldots, x_n\}$ is SOME solution for $\{x_{j'}, \ldots, x_n\}$

=> $O(j', n) \leq O(k, n)$ (WHY?)

=>

$$\begin{aligned} \mathrm{G}(1,n) &= 1 + \mathrm{G}(j', n) \\ &= 1 + \mathrm{O}(j', n) \quad \leftarrow \text{Induction hypothesis!} \\ &\leq 1 + \mathrm{O}(k, n) \\ &= \mathrm{O}(1, n) \quad \leftarrow \text{Final Result!} \end{aligned}$$

GY 3
Greedy Driving

Question

(CLRS-16.2-4) Professor Midas drives an automobile from Newark to Reno along Interstate 80.

His car's gas tank, when full, holds enough gas to travel m miles, and his map gives the distance between gas stations on his route. The professor wishes to make as few gas stops as possible along the way.

Give an efficient method by which Professor Midas can determine at which gas stations he should stop and prove that your algorithm yields an optimal solution.

Solution (i)

We prove that the simple greedy algorithm is optimal.
This is to drive to the furthest possible city that one can reach with the current gas in the car, fill up the tank and then continue, until reaching x_{n-1}.

Input to the problem is m and locations $x_0 < x_1 < \cdots < x_{n-1}$.
x_0 is the starting location and x_{n-1} is destination.

Note that a sequence $S = s_1, s_2, \ldots, s_k$ is a solution if

$$(i)\ 0 = s_0 < s_1 < s_2 < \cdots s_k < s_{k+1} = n - 1$$

$$(ii)\ \forall i \leq k, \qquad x_{s_{i+1}} - x_{s_i} \leq m$$

$|S| = k$ denotes the size of S.

The problem is to find the smallest (optimal) solution.

Solution (ii)

Consider the input X on n points: $x_0 < x_1 < \cdots < x_{n-1}$

G denotes greedy solution; O denotes an optimal solution.

Let g_i and o_i be the stops the two algorithms make, in order.

$$G = g_1, g_2, \dots, g_k \quad \text{and} \quad O = o_1, o_2, \dots, o_{k'}$$

Observation 1: $g_2, \dots, g_k$ is the Greedy solution for input $x_{g_1}, x_{g_1+1}, \dots, x_{n-1}$.

To see this, note that Greedy can be viewed as a recursive algorithm. After filling up at the first chosen station, Greedy runs itself starting from that station.

Solution (iii)

Consider the input X on n points: $x_0 < x_1 < \cdots < x_{n-1}$

G denotes greedy solution; O denotes an optimal solution.

Let g_i and o_i be the stops the two algorithms make, in order.

$$G = g_1, g_2, \ldots, g_k \quad \text{and} \quad O = o_1, o_2, \ldots, o_{k'}$$

Observation 1: $g_2, \ldots, g_k$ is the Greedy solution for input $x_{g_1}, x_{g_1+1}, \ldots, x_{n-1}$.

Observation 2: $O' = g_1, o_2, o_3, \ldots, o_{k'}$ is also an optimal solution for X.

(*) By the definition of Greedy, $o_1 \leq g_1$.

$g_1 = \max\{i : x_i - x_0 \leq m\}$
$x_{o_i} - x_0 \leq m$

$\Rightarrow$ $x_{o_2} - x_{g_1} \leq x_{o_2} - x_{o_1} \leq m$

$\Rightarrow$ O' is a solution for X

$\Rightarrow$ Since $|O'| = k' = |O|$, O' is also optimal.

Solution (iv)

Consider the input X on n points: $x_0 < x_1 < \cdots < x_{n-1}$

G denotes greedy solution; O denotes an optimal solution.

Let g_i and o_i be the stops the two algorithms make, in order.

$$G = g_1, g_2, \ldots, g_k \quad \text{and} \quad O = o_1, o_2, \ldots, o_{k'}$$

Observation 1: $g_2, \ldots, g_k$ is the Greedy solution for input $x_{g_1}, x_{g_1+1}, \ldots, x_{n-1}$.

Observation 2: $O' = g_1, o_2, o_3, \ldots, o_{k'}$ is also an optimal solution for X.

Observation 3': If $S = i_1, i_2, i_3 \ldots, i_t$ is an optimal solution for X then $S' = i_2, i_3 \ldots, i_t$ is an optimal solution for $X' = x_{i_1}, x_{i_1+1}, \ldots, x_{n-1}$.

First note that S' is a solution for X' since $\forall j,\ x_{i_{j+1}} - x_{i_j} \leq m$.

Suppose S' was not optimal for X'. Then X' would have another solution $U' = u_1, u_2 \ldots, u_{t'}$ with $t' < t - 1$.

Since $x_{u_1} - x_{i_1} \leq m$, $S'' = i_1, u_1, u_2 \ldots, u_{t'}$ would be a length $t' + 1 < t$ solution for X, contradicting the optimality of S! $\Rightarrow$ S' is optimal.

Solution (iv)

Consider the input X on n points: $x_0 < x_1 < \cdots < x_{n-1}$

G denotes greedy solution; O denotes an optimal solution.

Let g_i and o_i be the stops the two algorithms make, in order.

$$G = g_1, g_2, \ldots, g_k \quad \text{and} \quad O = o_1, o_2, \ldots, o_{k'}$$

Observation 1: $g_2, \ldots, g_k$ is the Greedy solution for input $x_{g_1}, x_{g_1+1}, \ldots, x_{n-1}$.

Observation 2: $O' = g_1, o_2, o_3, \ldots, o_{k'}$ is also an optimal solution for X.

Observation 3': If $S = i_1, i_2, i_3 \ldots, i_t$ is an optimal solution for X then $S' = i_2, i_3 \ldots, i_t$ is an optimal solution for $X' = x_{i_1}, x_{i_1+1}, \ldots, x_{n-1}$.

Observation 3: $O'' = o_2, o_3, \ldots, o_{k'}$ is an optimal solution for $x_{g_1}, x_{g_1+1}, \ldots, x_{n-1}$

Combine Observations 2 and 3'.

Solution (v)

Consider the input X on n points: $x_0 < x_1 < \cdots < x_{n-1}$

G denotes greedy solution; O denotes an optimal solution.

Let g_i and o_i be the stops the two algorithms make, in order.

$$G = g_1, g_2, \dots, g_k \quad \text{and} \quad O = o_1, o_2, \dots, o_{k'}$$

Observation 1: $g_2, \dots, g_k$ is the Greedy solution for input $x_{g_1}, x_{g_1+1}, \dots, x_{n-1}$.

Observation 2: $O' = g_1, o_2, o_3, \dots, o_{k'}$ is also an optimal solution for X.

Observation 3: $o_2, o_3, \dots, o_{k'}$ is an optimal solution for $x_{g_1}, x_{g_1+1}, \dots, x_{n-1}$.

We will prove optimality of Greedy by induction on n.

We assume that Greedy is optimal on all problems on set size $< n$.

(Basis: this is obviously true on sets of size 1 and 2.)

Solution (vi)

Consider the input X on n points: $x_0 < x_1 < \cdots < x_{n-1}$

G denotes greedy solution; O denotes an optimal solution.

Let g_i and o_i be the stops the two algorithms make, in order.

$$G = g_1, g_2, \dots, g_k \quad \text{and} \quad O = o_1, o_2, \dots, o_{k'}$$

Observation 1: $g_2, \dots, g_k$ is the Greedy solution for input $x_{g_1}, x_{g_1+1}, \dots, x_{n-1}$.

Observation 2: $O' = g_1, o_2, o_3, \dots, o_{k'}$ is also an optimal solution for X.

Observation 3: $o_2, o_3, \dots, o_{k'}$ is an optimal solution for $x_{g_1}, x_{g_1+1}, \dots, x_{n-1}$.

We will prove optimality of Greedy by induction on n.

We assume that Greedy is optimal on all problems on set size $< n$.

Now consider any problem of size n.
G is Greedy solution; O is any optimal solution.

Solution (vii)

We assume that Greedy is optimal on all problems on set size $< n$.

$$G = g_1, g_2, \dots, g_k \quad \text{and} \quad O = o_1, o_2, \dots, o_{k'}$$

Observation 1: $g_2, \dots, g_k$ is the Greedy solution for input $x_{g_1}, x_{g_1+1}, \dots, x_{n-1}$.

Observation 2: $O' = g_1, o_2, o_3, \dots, o_{k'}$ is also an optimal solution for X.

Observation 3: $o_2, o_3, \dots, o_{k'}$ is an optimal solution for $x_{g_1}, x_{g_1+1}, \dots, x_{n-1}$.

Observation 4: $g_2, \dots, g_k$ is an optimal solution for $x_{g_1}, x_{g_1+1}, \dots, x_{n-1}$

From Observation 1 and induction hypothesis on Greedy correctness.

Observation 5: $k - 1 = k' - 1$

From Observations 3 and 4.

$\Rightarrow\ k = k' \ \Rightarrow$ **greedy is optimal for our original set!**