COMP3821/COMP9801 Workshop Week 2

Greedy Algorithms

Xueyi Chee and Jeremy Le

2025T3

Announcements

- Problem Set 1 Submission due on Friday Week 2.
  - Submissions on Gradescope.
- Project
  - Project mentor consultations have begun this week.
  - Complete Group Formation Form.
  - Proposal due Sunday Week 3.
- Help sessions on Friday 11am - 1pm for Weeks 1 - 10.

Guided Problem 1

Guided Problem

Let $\mathcal{I} = \{I_1, \ldots, I_n\}$ be a set of n intervals on the real line; the ith interval can be described by a pair of real numbers a_i, b_i such that interval $I_i = [a_i, b_i]$. A *clique cover* is a partitioning of the intervals $\{C_1, \ldots, C_k\}$ such that, whenever two intervals are in the same set C_i, they intersect in at least one point.

Describe and analyse a polynomial-time algorithm that computes the smallest clique cover of $\mathcal{I}$.

Guided Problem 1 Example

Algorithm: Sort by ascending b_i values

everything intersects the earliest uncovered b_i forms a clique

- keep going until all intervals are covered.

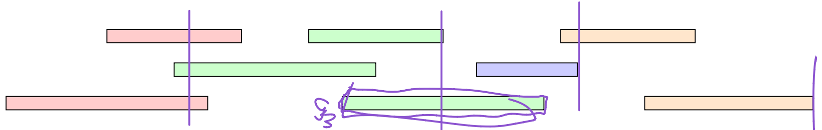

Figure: *The minimum clique cover size is four.*

$\boxed{i=3}$

$$G = \{g_1, g_2, \dots, g_k\}$$

$$O = \{\boxed{o_1, o_2,} \cancel{\dots} o_k\}$$

o_7

$g_i \neq o_i$

Guided Problem 1 Working

Claim 1: There exists an optimal clique cover $O = \{C_1, \ldots, C_k\}$ such that each clique C_i intersects at the right endpoint of some interval $i \in C_i$

Suppose you have some C_i which did not "generate" from some right endpoint

Shift that intersection point to the first right endpoint

All intervals will still be covered.

Guided Problem 1 Working

Claim 2: The set of cliques produced by the greedy algorithm is optimal.

Let $S = \{C_1, \dots, C_k\}$ and $O = \{O_1, \dots, O_\ell\}$ optimal
(greedy)

By Claim 1 - each clique generated by some right endpoint.

Denote x_i endpoint for each C_i - ~~gen~~ intersection point

y_i " O_i "

Assume $x_1 \leq \dots \leq x_k$ and $y_1 \leq \dots \leq y_\ell$.

Guided Problem 1 Working

If $S \subseteq O$ $\quad S = O$ then S is optimal

$|S| < |O|$ contradiction

$S \not\subseteq O$

Less formal $\quad C_i = O_i$ up until some i

- let C_j earliest clique in S not in O
- let I_j denote the interval which corresponds to containing x_j.

Guided Problem 1 Working

Claim 2a: For each $j' \in \{1, \ldots, \ell\}$, define $O'_{j'} := O_{j'} \setminus C_j$.

Then $O' := \{O'_1, \ldots, O'_\ell\} \cup \{C_j\}$ is a $\underbrace{\text{feasible solution}}$

and that $C_i \in O'$ for all $i \leq j$.

① minimality

② clique cover

Guided Problem 1 Working

Claim 2aa: O' is minimal in size

O' covers the entire interval

$|O| \leq |O'|$ because O is optimal

Claim:

let $O_j' \in O$ that contains I_j. $O_j' \subseteq C_j$

let $I \in O_j'$, $I \cap I_j \neq \emptyset$

$\Rightarrow a_I \leq b_j$

I not in any other previous clique

then $I \in C_j$

Guided Problem 1 Working

This $O_{j'} = \emptyset$

$|O'| \leq (|O| - 1) + 1 = |O|$.

Guided Problem 1 Working

Claim 2ab: O' is a feasible clique cover.

O' is a set of cliques because $O_i \setminus \{I\}$ is still a clique

$O_{j'}' := O_{j'} \setminus C_j$

$O' = \underbrace{\{ O_1', \ldots, O_\ell' \}} \cup \{ C_j \}$

Clearly ~~...~~ C_j disjoint with every O_i'

O_i' and O_j' → disjoint since O was a feasible solution.

$\in O$

original O_i and O_j are disjoint

Guided Problem 1 Working

Claim 2b: $C_i \in O'$ for all $i \leq j$

$j-1$ are equal by choice of j

$C_j \in O'$ union ~~so~~ at the end

Guided Problem 1 Working

Conclusion: O' contains the first j digits of

S

repeat this for all points of difference

Time Complexity: $O(n\log n)$ sorting

$O(n)$ sweep

Guided Problem 1 Solution Overview

- *Algorithm:* Sort intervals in increasing order of b_i. Take the earliest b_i and the set of intervals that intersect this point are included in the clique. Remove all covered intervals and repeat until all intervals are in a clique.
- *Intersections at right endpoint*. Let $O = \{C_1, \ldots, C_k\}$ denote an optimal solution. If all intervals in C_i, intersect at a non-right endpoint of any interval in C_i, we can move towards the right until it hits the right endpoint of an interval. All intervals in C_i, must still intersect this point.
- This makes it sufficient to consider optimal solutions that generate cliques using right endpoints.

Guided Problem 1 Solution Overview

Exchange Preliminaries

- Let $S = \{C_1, \ldots, C_k\}$ and $O = \{O_1, \ldots, O_\ell\}$ denote set of cliques from greedy and optimal solution respectively.
- From previous slide, we may assume intersection points occur at right endpoint of an interval in each clique.
- For each clique $C_i \in S$, let x_i denote the earliest right endpoint among all intervals in C_i. Use a similar notation for $O_j \in O$ with y_j denoting the earliest right endpoint in O_j.
- Without loss of generality, we assume our solutions are ordered such that $x_1 \leq \cdots \leq x_k$ and $y_1 \leq \cdots \leq y_\ell$.
- If $S \subseteq O$, then since O optimal, we either have $S = O$ or we can extend S. Thus, we can assume that $S \not\subseteq O$.
- Then there exists $C_j \in S$ such that $C_j \notin O$. Let C_j be the earliest such clique and $I_j \in C_j$ be the interval such that the right endpoint of I_j is x_j.

Guided Problem 1 Solution Overview

- For each $j' \in \{1, \ldots, \ell\}$, define $O'_{j'} := O_{j'} \setminus C_j$.
- We claim that $O' := \{O'_1, \ldots, O'_\ell\} \cup \{C_j\}$ is a feasible solution and that $C_i \in O'$ for all $i \leq j$.
- We first prove this by showing that O' has the same number of cliques as O (optimal in size) then show that O' is a proper clique cover.

Guided Problem 1 Solution Overview

Size Optimality

- Clearly, O' covers the entire interval set and note that $|O| \leq |O'|$ since O' is a feasible clique cover.
- Now, let $O_{j'}$ in O that contains I_j. We claim $O_{j'} \subseteq C_j$.
  - Let $I \in O_{j'}$, then $I \cap I_j \neq \emptyset$ which implies $a_I \leq b_j$ and since I is not contained in any previous clique, so $I \in C_j$.
- Therefore, $O'_{j'} = \emptyset$, so, $|O'| \leq (|O| - 1) + 1 = |O|$.
- Hence O' has the same number of cliques as O.

Guided Problem 1 Solution Overview

Proof of Clique Cover

- Each O'_i is a clique since $O_i \setminus \{I\}$ is still a clique for any $I \in \mathcal{I}$. Therefore, O' is a set of cliques.
- We now prove O' forms a set of disjoint cliques.
  - First, O'_i and C_k are disjoint for all i.
  - For any pair of distinct indices i, i', we have that O'_i and $O'_{i'}$ are disjoint from the feasibility O.
- Finally, we show $C_i \in O'$ for all $i \leq j$.
  - Clearly, $C_j \in O'$. Then since C_j is earliest point of difference, then $C_i \in O'$ for all $i < j$.
  - Since O is a feasible solution, it follows that $C_i \cap C_j = \emptyset$ and so $C_i \in O'$.

Guided Problem 1 Solution Overview

- *Exchange Conclusion:* We have shown there exists an optimal solution that contains the first j cliques of S. We can repeat the same argument to show that there exists an optimal clique cover that contains S. This implies S is optimal.
- *Time Complexity:* $O(n \log n)$ as merge sort intervals in $O(n \log n)$ and traverse intervals in $O(n)$ time.

Guided Problem 2

Guided Problem

Your friends are starting a security company that needs to obtain licenses for n different pieces of cryptographic software. Due to regulations, they can only obtain these licenses at the rate of at most one per month.

Each license is currently selling for a price of \$100. However, they are all becoming more expensive according to exponential growth curves: in particular, the cost of license j increases by a factor of $r_j > 1$ each month, where r_j is a given parameter. This means that if license j is purchased t months from now, it will cost $100 \cdot r_j^t$. We will assume that all the price growth rates are distinct; that is, $r_i \neq r_j$ for licenses $i \neq j$ (even though they start at the same price of \$100).

Given that the company can only buy at most one license a month, describe and analyse a polynomial-time algorithm that takes the n rates of price growth $r_1, \ldots, r_n$ and computes an order in which to buy the licenses so that the total amount of money spent is minimised.

Guided Problem 2 Example

100

$100(5^2)$

i	r_i
1	5
2	2
3	3

100
100

100

Buying Sequence: 3, 2, 1

Cost: $100 + 100(2) + 100(5^2)$

Guided Problem 2 Working

prove buying high rate to low rate minimises total cost

$L_1, L_2, L_3 \quad \dots \quad L_n$

$r_1 > r_2 > r_3 > \dots > r_n$

$$\boxed{r_i < r_{i+1}}$$

Guided Problem 2 Solution Overview

$r_i < r_{i+1}$

$$\text{Cost}_b = \boxed{100 + 100 r_1 + 100 r_2^2 + \cdots} + 100 r_i^i + 100 r_{i+1}^{i+1} + \cdots$$

$$\text{Cost}_a = \boxed{100 + 100 r_1 + 100 r_2^2 + \cdots} + 100 r_{i+1}^i + 100 r_i^{i+1} + \cdots$$

$$\Delta \text{cost} = \text{Cost}_a - \text{Cost}_b$$

$$= 100 r_{i+1}^i + 100 r_i^{i+1} - 100 r_i^i - 100 r_{i+1}^{i+1}$$

$$= 100 \left[r_{i+1}^i + r_i^{i+1} - r_i^i - r_{i+1}^{i+1} \right]$$

$$= 100 \left[-r_{i+1}^i [1 - r_{i+1}] + r_i^i [r_i - 1] \right]$$

Guided Problem 2 Solution Overview

$$= 100 \Big[\underbrace{\underbrace{r_i^{\,i}[r_i - 1]}_{} - \underbrace{r_{i+1}^{\,i}[r_{i+1} - 1]}_{}}_{<0} \Big)$$

$=$

$$f(r) = r^i(r-1)$$

$$r_i < r_{i+1}$$

$$f(r_i) < f(r_{i+1})$$

$$r_i^{\,i}(r_i - 1) < \underbrace{r_{i+1}^{\,i}}_{}(r_{i+1} - 1)$$

Guided Problem 2 Solution Overview

Δ cost is negative

$\therefore O(n \log n)$