Summary Part

1. **In your own words, explain what a greedy algorithm is and how they work.**

   A greedy algorithm is a simple, intuitive algorithmic approach that makes the best possible choice at each step with the hope of finding a global optimum. It follows the problem-solving heuristic of making the locally optimal choice at each stage with the goal of reaching a global optimum.

2. **What is a power set? How can you compute it?**

   A power set is the set of all subsets of a given set, including the empty set and the set itself. To compute the power set of a set with n elements, the generate all combinations of these elements, resulting in 2^n subsets.

3. **What is the time complexity of problems like the set-covering problem?**

   The set-covering problem is a classic example of an NP-hard problem, meaning its time complexity is exponential in the worst case. Specifically, the best known algorithms have time complexities that are not polynomial, typically $O(2^n)$ or similar, making them impractical for large datasets.

4. **What makes "Approximation Algorithms" useful?**

   Approximation algorithms are useful because they provide solutions to NP-hard problems within a factor of the optimal solution in polynomial time. This makes them practical for real-world applications where exact solutions are computationally infeasible.

Self-reflection part

What did you learn?

I learned about the fundamentals and applications of greedy algorithms, the concept and computation of power sets, the complexity of the set-covering problem, and the importance of approximation algorithms in solving computationally hard problems.

What went smoothly?

Understanding the greedy algorithm and the power set concepts went smoothly

What was difficult about the content this week?

Grasping the time complexity implications of NP-hard problems and the intricacies of approximation algorithms was challenging, especially when it came to understanding why certain problems cannot be solved efficiently.

How will you approach things differently next time?

I will focus more on understanding the foundational theories behind NP-hard problems

Do you have any feedback about the content for this week?

Good!