There are 2 questions for a total of 10 points.

---

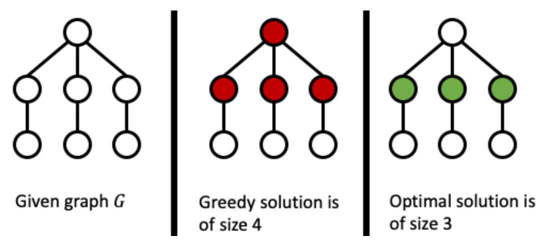
1. (3 points) Consider the following graph problem:

   Given an undirected, graph $G = (V, E)$ find the smallest subset $S \subseteq V$ of vertices such that for every edge $(u, v) \in E$, at least one of u, v belongs to S.

   Construct a counterexample to show that the following greedy algorithm does not work for this problem.

   GreedySet(G)
   - While G has at least one edge:
     - Let v be the vertex in G with maximum degree (ties are broken arbitrarily)
     - $S \leftarrow S \cup \{v\}$
     - Remove v and its edges from G to obtain G'
     - $G \leftarrow G'$
   - return(S)

   Solution: Here is a counterexample.

   You need not draw. The same graph can be represented as $G = (V, E)$ where $V = \{1, 2, 3, 4, 5, 6, 7\}$ and $E = \{(1, 2), (1, 3), (1, 4), (2, 5), (3, 6), (4, 7)\}$. The greedy algorithm picks vertex 1 which forces it to pick 3 other vertices while the optimal solution is $\{2, 3, 4\}$.

2. Suppose you are running a trucking company that ships packages from San Diego (SD) to Los Angeles (LA). The amount of packages requires you to send a number of trucks each day from SD to LA. Each truck has a limit of W on the maximum amount of weight they are allowed to carry. Boxes arrive in LA one by one, and each package i has weight w_i. The trucking station is quite small so at most one truck can be at the station at any time. Company policy requires that boxes are shipped in the order they arrive; otherwise, a customer might get upset upon seeing a box that arrived after this customer's box, make it to LA faster. The company is using the following greedy strategy:

   Greedy Strategy: Pack boxes in the order they arrive and whenever the next box does not fit, send the truck on its way and repeat on the next truck.

   (a) Fill in the steps of the proof of the following modify-the-solution lemma:

   Lemma: *For any input $I = (w_1, \ldots, w_n; W)$, let $[1, \ldots, g]$ be the packages loaded into the first truck by the greedy algorithm. Let OS be a solution that does not load packages $[1, ..., g]$ onto the first truck. Then there is a solution OS' that loads packages $[1, ..., g]$ onto the first truck and uses the same number of trucks as OS.*

   Proof: Let OS be as above.

i. (1 ½ points) Define OS'.

> **Solution:** Let the first truck contains packages $[1, \ldots, i']$ in OS. By the nature of the greedy choice, $[1, \ldots, g]$ is the maximum number of packages able to fit in the first truck so this implies that $i' < g$. Create OS' by moving packages $[i' + 1, \ldots, g]$ from the next few trucks onto the first truck.

ii. (1 ½ points) OS' is a valid solution because... (Justify why in OS', no truck is overloaded.)

> **Solution:** We must show that there is no truck that is overloaded. We know that $[1, \ldots, g]$ can fit in the first truck and by the validity of OS, none of the other trucks were overloaded to begin with. Then we only took packages off trucks and so in OS', they still are not overloaded.

iii. (1 point) The number of trucks in OS' is less than or equal to that in OS because... (justify).

> **Solution:** We did not add any more trucks. Therefore, $|OS| = |OS'|$.

(b) (2 points) Fill in the steps of the inductive argument.
<u>Claim:</u> *For any input of any size $n \geq 1$, the greedy solution is optimal.*
Base Case: For $n = 1$, put the package on the first truck and it is the optimal solution.
Inductive Hypothesis: Suppose that for some $n > 1$, the greedy strategy is optimal for all inputs of size k such that $1 \leq k < n$.
Inductive Step: Let $I = (w_1, \ldots, w_n; W)$ be any input instance of size n. Let $[1, ..., g]$ be the packages loaded onto the first truck by greedy solution $GS(I)$. Let $I' = (w_{g+1}, \ldots, w_n; W)$. Let OS be any solution of I. Then by the exchange argument, there exists a solution OS' that fills the first truck with packages $[1, \ldots, g]$ and uses at most as many trucks as OS. Note that $|GS(I)| = 1 + |GS(I')|$ and $|OS'| = 1 + |S(I')|$, where $S(I')$ is some solution for instance I'. Therefore we have... (*complete the argument*)

> **Solution:**
>
> $$\begin{aligned} |GS(I)| &= 1 + |GS(I')| \\ &\leq 1 + |S(I')| \quad \text{(using induction hypothesis since } I' \text{ has less than } n \text{ packages)} \\ &= |OS'| \\ &\leq |OS| \quad \text{(using the exchange lemma)} \end{aligned}$$
>
> This shows that the greedy solution is as good as any other solution and this completes the inductive step.

(c) (1 point) Here is pseudo-code for an efficient implementation of the greedy strategy. Give the running time expression for this algorithm as a function of n in big-O notation.

> `ScheduleTrucks`$(w_1, \ldots, w_n; W)$
> - output = [ ]
> - $i = 1$
> - $start = 1$
> - **while** $i \leq n$:
>   - $sum = 0$
>   - **while** $sum < W$:
>   - $sum = sum + w_i$

        - $i = i + 1$.
    - output = output $\cup$ [start,$i - 1$]
    - start = i
- **return** output

Solution: The running time is $O(n)$.