Homework 3: Greedy Algorithms

Due: October 01, 2024

This homework must be typed in LaTeX and submitted via Gradescope.

Please ensure that your solutions are complete, concise, and communicated clearly. Use full sentences and plan your presentation before your write. Except where indicated, consider every problem as asking for a proof.

Problem 1. Recall that a set of vectors $\{v_1, v_2, \ldots, v_n\}$ is called **linearly independent** if the only solution to the equation

$$c_1 v_1 + c_2 v_2 + \cdots + c_n v_n = 0$$

is $c_1 = c_2 = \cdots = c_n = 0$. In other words, the vectors are linearly independent if no vector in the set can be written as a linear combination of the others.
A **spanning set** is a collection of vectors that can be combined through linear combinations to generate every element of the vector space.
Finally, a **basis** is a linearly independent set that spans the vector space. This implies that every vector in the space can be expressed uniquely as a linear combination of the basis vectors.

Let M be an $n \times m$ matrix. Let R be the set of rows of M. Let ℓ be a subset of $\mathcal{P}(R)$ such that

$$\ell = \{A : A \subset R, A \text{ linearly independent}\}.$$

(a) Show that (R, ℓ) is a matroid.

Hint: It might be helpful to consider components.

(b) Let $w : \ell \to \mathbb{R}^+$ be a function assigns to each subset $A \subset R$ its cardinality, i.e., $w(A) = |A|$. To be consistent with the lecture notes, we also define w as a function of R: $w(v) = 1$ for every $v \in R$. Observe that $w(A) = |A| = \sum_{v \in A} w(v)$.

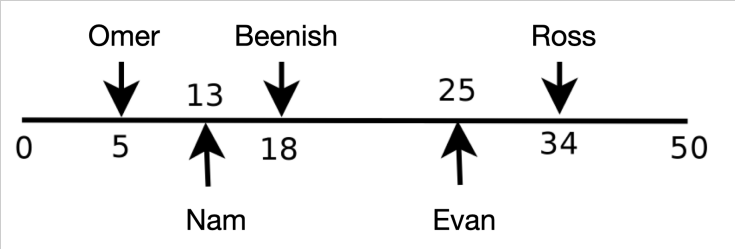
Let IsLinearlyIndependent be a predicate that checks whether a subset of R is linearly independent or not. Assume this algorithm runs in $O(T)$ time.

Design an efficient algorithm to find a basis of the vector space spanned by R. Prove the correctness/optimality of your algorithm and analyze an asymptotically tight runtime.

Solution. □

Problem 2. The power lines along a country road have been modified to carry broadband Internet. Wi-Fi towers are being built along the road to provide the community with internet. To find the minimum number of towers required so that each house is sufficiently close to at least one tower, we model the problem as follows:

a) The entire course staff has taken up residence on Algorithm Street. The diagram below shows where they all live on the street.



Omer lives 5 miles down the road, Nam lives 13 miles down the road, and so on. Wi-Fi towers have an effective radius of 5 miles. Determine the minimum number of Wi-Fi towers needed such that each staff member has internet, and give the locations for these towers as well.

b) We're given a line segment ℓ, a set of non-negative numbers N that represents the locations of customers on ℓ, and a distance d. We wish to find a set of Wi-Fi towers of minimal size on ℓ such that each location in N is at most d away from some tower. Give an efficient greedy algorithm that returns a minimum size set of points. Prove its correctness and justify its runtime.

Now we generalize our model to account for houses that are not by the side of the road.

c) We're given a line segment ℓ, a set of pairs N representing the locations of customers, and a distance d. For each pair $(x, y) \in N$, let $x \in [0, \infty)$ be the distance along ℓ and $y \in [-d, d]$ be the distance above or below ℓ. We wish to find a set of Wi-Fi towers of minimal size on ℓ such that each location in N is at most distance d from some tower (here, we are using Euclidean distance).

Modify your algorithm from part (b) to solve this variation of the problem. You do not need to prove its correctness, but please explain how your proof from part (b) would (or would not) need to change based on your modifications.

Can we generalize further?

d) Does the correctness of your algorithm depend on the fact that ℓ is a line segment and not some curve? If so, give an example that illustrates the problem with your algorithm when ℓ is a curve. If not, explain how your algorithm could handle a curve. You shouldn't be writing another algorithm, or modifying your existing algorithm, just explain your reasoning.

Solution. □