

CS 161 W22: Section 8 Problems

February 2022

Exercise 0

Sometimes it can be tricky to tell when a greedy algorithm applies. For each problem, say whether or not the greedy solution would work. If it wouldn't, give a counter example.

1. You have unlimited objects of different sizes, and you want to completely fill a bag with as few objects as possible. (Greedy: Keep putting in the largest object possible given the space you have left.)
2. You have unlimited objects, all of which are size 3^k for different integers k , and you want to completely fill a bag with as few objects as possible. (Greedy: Same approach as the previous part.)
3. You have lines that can fit a fixed number of characters. You want to print out a given piece of text while using as few lines as possible. (Greedy: Always fit as many words as you can on the next line.)

Exercise 1

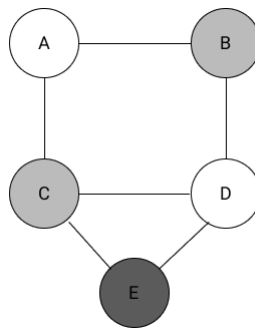
Suppose we are given n ropes of different lengths, and we want to tie these ropes into a single rope. The cost to connect two ropes is equal to sum of their lengths. We want to connect all the ropes at minimum cost.

For example, suppose we have 4 ropes of lengths 7, 3, 5, and 1. One (not optimal!) solution would be to combine the 7 and 3 rope for a rope of size 10, then combine this new size 10 rope with the size 5 rope for a rope of size 15, then combine the rope of size 15 with the rope of size 1 for a final rope of size 16. The total cost would be $10 + 15 + 16 = 41$. (Note: the optimal cost for this problem is 29. How might you combine the ropes to achieve that cost?)

Find a greedy algorithm for the minimum cost and prove the correctness of your algorithm.

Exercise 2

Minimum graph coloring is an open NP-hard problem for finding the minimum number of colors needed to color all the nodes in a graph such that no nodes of the same color share an edge.



1. Although the problem is NP-hard, we can use greedy algorithms to obtain a pretty good solution. Describe a greedy algorithm that never uses more than $d + 1$ colors, where d is the maximum degree of a vertex in the given graph. Your algorithm should run in $O(n^2)$ where n is the number of nodes.

2. Prove by counterexample that your greedy algorithm does not always return the correct minimum coloring. Your solution should include a graph, the correct minimum coloring, and the coloring returned by the greedy algorithm.
3. Prove that your greedy algorithm will return a coloring that uses at most $d + 1$ colors. (Note: You may use proof by induction, but you do not need to for this problem.)