CSCI 432 Handout 11: Greedy Algorithms

Name: _____

March 26, 2026

Class Scheduling

Consider the problem of class scheduling. There are n classes available, and you want to take the maximum number of classes. Each class has a start time and a finish time:

$$C = \{c_i = (s_i, f_i)\}_{i=1}^{n}.$$

What strategies allow you to take the most classes? For this, we will consider several different strategies. We will call a strategy optimal if it is one that maximizes the number of classes that you can take.

For the strategies listed below, are they optimal or not? If not, provide a counter-example to prove that it is not optimal.

1. Starts first: Choose the class that starts first, eliminate any conflicts with that. Among the remaining, choose the class that starts first and repeat this process until no classes are remaining.

   Answer

2. Starts last.

   Answer

3. Finishes first.

   Answer

4. Finishes last.

   Answer

5. Longest duration.

   Answer

6. Shortest duration.

   Answer

7. Bonus: Randomized selection (here, find an example that shows where random selection will have a low probability of finding an optimal solution).

   Answer

Hoffman Codes

Consider the following code (generated using CatChat[1]:

Table 1: Prefix-free binary code for the English alphabet

Letter	Codeword
A	00000
B	11111000
C	111100
D	1111101
E	01100
F	1111010
G	1111011
H	1100
I	0010
J	111111100
K	111111101
L	111001
M	11111001
N	0100
O	00010
P	0011
Q	0111
R	1101
S	111010
T	0101
U	111000
V	00001
W	111011
X	01101
Y	00011
Z	11111111

3. Can you decode the following message:
   01100111110010000000101110011111100101100111101000010110111111000000100100111000111010
   Answer

[1]Prompt was "latex table with a prefix-free code for the alphabet." The answer had a formatting problem, but was easily fixable. However, the code given was not prefix free, so I had to fix that.

4. For this code, draw a binary tree, withe the encoded characters stored at the leaf nodes.

   Answer

5. For the Huffman strategy of "Merge the two least frequent letters and recurse", draw a the Huffman tree for the following frequency count:

Table 2: Random frequency count for letters in an imaginary text that loves weird letters.

Letter	**Codeword**
A	7
B	13
C	2
D	15
E	4
F	12
G	0
H	9
I	5
J	11
K	1
L	14
M	3
N	10
O	6
P	8
Q	15
R	2
S	13
T	4
U	9
V	0
W	12
X	5
Y	7
Z	11