

Name: \_\_\_\_\_

1. (1 pt.)

- **Read all material carefully.**
- *If in doubt whether something is allowed, ask, don't assume.*
- You may refer to your **books, papers, and notes** during this test.
- **E-books** may be used *subject to the restrictions* noted in class.
- **Computers** (including smart phones, tablets, etc.) **are not permitted**, except when used strictly as e-books or for viewing ones own notes.
- Calculators are not required but are permitted.
- **Network access** of any kind (cell, voice, text, data, ...) is **not permitted**.
- Write, and draw, carefully. **Ambiguous or cryptic answers receive zero credit.**
- Use **class and textbook conventions** for notation, algorithmic options, etc.
- **Do not attach or remove any pages.**

**Write your name** in the space provided above.

**Do not write anything else on this page.**

WAIT UNTIL INSTRUCTED TO CONTINUE TO REMAINING QUESTIONS.
---

(Do not view any other pages.)

**Do not write on this page.**  
(It is for use in grading only.)

Q	Full Score
1	1
2	5
3	5
4	5
5	5
6	5
7	5
8	5
9	9
total	45

2. (5 pts.) Consider the following binary-string code, following the conventions used in class:  $(a, b, c) = (01, 00, x)$ . List **all** possible values for  $x$  that result in a *prefix-free code*. **Explain your answer.**

3. (5 pts.) Repeat Question 2 but with *prefix-free code* replaced by *fixed-length code*

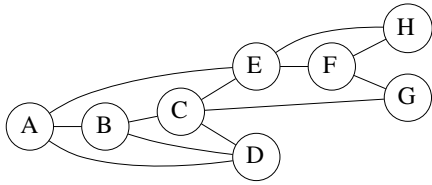
4. (5 pts.) Following the conventions used in class and the textbook, depict the **final tree** produced by *Huffman's Algorithm* on the following input. (You do *not* need to depict intermediate trees, but doing so will permit better partial credit in case of errors.)

character:	C	D	F	T	V	W
frequency:	39	17	22	30	6	5

5. (5 pts.) Describe as **precisely** as you can, in plain English (or, if you prefer, pseudocode), the *standard greedy algorithm* for the (coin) *change-making problem* as discussed in class. Be sure to clearly indicate the input and output in addition to the program logic.

6. (5 pts.) Is the algorithm of Question 5 guaranteed to provide an optimal solution? **Explain your answer** as precisely as you can, providing either proof of or counterexample for the claim of optimality.

7. (5 pts.) Using the conventions described in class, depict the *adjacency lists* representation of the following graph.



8. (5 pts.) Repeat Question 7 using *adjacency matrix* instead of *adjacency lists*.

9. (9 pts.) Trace the action of the *furthest-in-future* cache-replacement strategy on the following sequence of block requests, for a cache size of 4 blocks. Break ties by evicting the lower-numbered block. **Depict the state of the cache after each request** using the format `request: resulting-cache-contents`. State the **number of cache misses**.

9 6 0 5 1 8 7 0 7 2 1 1 3 4 9 9