RULES:

1. Don’t start the exam until the instructor says to.

2. This is a closed-book, closed-notes, no-calculator exam. Don’t refer to any materials other than the exam itself.

3. Write your name, and your login name, on each page of the exam when you get to it.

4. Do not look at anyone else’s exam. Do not talk to anyone but an exam proctor during the exam. If you’re wearing a billed cap, please turn it around or take it off. Turn off cell phones and pagers and MP3 players.

5. If you have a question, raise your hand and an exam proctor will come to you.

6. You have two hours and thirty minutes to finish the exam. When you are done, give your exam to a proctor. The proctor will check your picture ID and sign the ID check below. Be sure you turn in all the pages! (The last 2 pages are blank and can be used as scratch paper.)

7. Your exam and course grade will be emailed to your account early next week. Exams will be returned next quarter, in the CSE student affairs offices.

<table>
<thead>
<tr>
<th>#</th>
<th>Type</th>
<th>max pts</th>
<th>actual pts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>General TF</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>General matching</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>General numbers</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>Coding</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>Graphs</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Treaps</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>Hashing</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td>140</td>
<td></td>
</tr>
</tbody>
</table>

grader:______

PROCTOR ID CHECK:________________
1. [26 pts.] (1 point for each correct; -1 point for each incorrect; 0 points each left blank) True or False:

a. ______ Skip lists and hash tables have the same big-O worst-case time cost for insert operations.

b. ______ Red-black trees and B-trees have the same big-O worst-case time cost for delete operations.

c. ______ A binary search tree with 10,000 nodes is a sparse graph.

d. ______ In a nonempty Huffman code tree, there must be an odd number of nodes.

e. ______ Hashtables are good for efficiently iterating over keys in sorted order.

f. ______ A B-tree is a binary tree.

g. ______ Any AVL rotation in a BST preserves the BST ordering property.

h. ______ An undirected graph is connected if for every pair of distinct vertices V1, V2, there is a path from V1 to V2.

i. ______ In a red-black tree, if a red node has two child nodes, those child nodes must be the same color.

j. ______ In an AVL tree, the levels of any two leaves can differ by at most 1.

k. ______ In a binary heap, the levels of any two leaves can differ by at most 1.

l. ______ The weighted shortest-path problem is NP-complete.

m. ______ Every subtree of an AVL tree is itself an AVL tree.

n. ______ Every subtree of a BST is itself a BST.

o. ______ Every subtree of a treap is itself a treap.

p. ______ Every subtree of a spanning tree for a graph G is a spanning tree for G.

q. ______ In a Huffman code, the code for a symbol is never a prefix of the code for another symbol.

r. ______ In a randomized search tree, the values of the priorities associated with keys that have been inserted determine which key is in the root.

s. ______ A precondition of Kruskal’s algorithm is that the input graph not have negative-weight edges.

t. ______ With double hashing, collisions never occur.

u. ______ Considered as an undirected graph, every tree with N nodes has the same number of edges.

v. ______ Closed hashing is also known as open addressing.

w. ______ Every minimum-cost spanning tree is a connected graph.

x. ______ Depth-first search is the best approach for finding shortest paths in unweighted graphs.

y. ______ Keys in a binary search tree must be comparable to each other.

z. ______ Randomized search trees are about as efficient in the average case as red-black trees, and are easier to implement.
2. [28 pts.] (2 pts ea) Fill in the blank with the number of the best answer from this list. Some may be used more than once and some may not be used at all.

1. Simple path  2. Priority  3. Heap
4. DAG  5. Binary trie  6. Amortized analysis
16. Queue  17. Parent pointers  18. Union by height
22. std::map  23. Stack  24. AVL tree
25. NP complete  26. Path compression  27. 2-3 tree

a. _______ Has the biggest difference between average-case and worst-case find operations.
b. _______ A Huffman code tree is one of these.
c. _______ Acts like a pointer in C++.
d. _______ Such problems are intractable.
e. _______ Store nodes here to do depth-first search.
f. _______ Usually has a fixed MAXLEVEL.
g. _______ Makes find operations almost constant time.
h. _______ Every tree is one of these.
i. _______ Simulates randomness.
j. _______ Space efficient for dense graphs.
k. _______ Red-black trees are used for this.
l. _______ Shows whether self-adjusting finds are worth it.
m. _______ Used in high-performance disk filesystems.
n. _______ Randomized search trees randomize this.
3. [27 pts] (3 pts each) Write down the correct (exact) number in the space provided.

   a. _____ The average-case number of key comparisons for an unsuccessful find in a completely filled binary search tree with 7 nodes.

   b. _____ The average-case number of key comparisons for a successful find in a linked list with 7 nodes.

   c. _____ The worst-case number of key comparisons for a successful find in an AVL tree with 4 nodes.

   d. _____ The worst-case number of key comparisons for an unsuccessful find in a randomized search tree with 7 nodes.

   e. _____ The best-case number of key comparisons for a successful find in a red-black tree with 7 nodes.

   f. _____ The minimum number of edges in a spanning tree for a connected undirected graph with 10 vertices.

   g. _____ The average-case number of key comparisons for a successful find in a binary search tree with 3 nodes (assuming all key insert orders are equally likely, and all keys are equally likely to be searched for).

   h. _____ The probability of a collision when inserting two keys in an initially empty hash table of length 10 using random hashing.

   i. _____ The number of bytes in a Huffman code for a message consisting of 8 distinct symbols of equal probability.
4. [18 pts total] You are given a representation of a Huffman code tree as an adjacency matrix $M$. In this representation, if a node has two children, the child with the smaller index is the “0” child of its parent. The dictionary $D$ associates each leaf node index with the alphabet symbol in that leaf node.

**M:**

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**D:**

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>6</th>
<th>7</th>
<th>1</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>P</td>
<td>R</td>
<td>L</td>
<td>S</td>
<td></td>
</tr>
</tbody>
</table>

a. [6 pts] In the space above, draw the Huffman code tree. Label all internal nodes with their index. Label leaves with the letter in that leaf node. Label links as 0 or 1 as appropriate.

b. [2 pts] According to this code, what are the two least common symbols? ____, ____

c. [4 pts] According to this code, write down the left-to-right sequence of bits that codes this message: SPELLER (The bits may not completely fill the 3 bytes shown.)

```
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
```

d. [6 pts] Write down the message coded by this sequence of bits: 0 1 1 1 1 1 0 0 0 1 0 1 1 0 1 1 0

_____ _____ _____ _____ _____ _____
5. [18 pts total] Consider the following adjacency list for an unweighted directed graph $G$:

```
1  →  2
2  →  4  →  3
3  →  5
4  →  5  →  3
5
6  →  1
```

a. [6 pts] A depth-first search starting from vertex 6 will visit vertices in $G$ in some sequence. In fact, three distinct sequences are possible in this case. Show all of them:

____, ____, ____, ____, ____, ____
____, ____, ____, ____, ____, ____
____, ____, _____, ____, _____, ____

b. [12 pts] Show the result of running Dijkstra’s algorithm on $G$, with start vertex 6. Assume all edges have cost (i.e. distance) 1. For each entry in the table, give the index of the previous vertex on the shortest path (or -1 if none), and the distance to the start vertex along that path.

<table>
<thead>
<tr>
<th>vertex</th>
<th>previous</th>
<th>distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6. [10 pts] Show the treap resulting from inserting the following (key,priority) pairs, in order shown, into an initially empty treap. (Keys here are characters, with alphabetic ordering; priorities are integers, with larger integers representing higher priority.) Label each node clearly with the (key,priority) pair in it.

(W,30), (P,0), (L,5), (K,70), (Z,60), (A,20)

7. [13 pts total] This hashtable is the result of a sequence of 5 insert operations. Linear probing (index increment) was used for collision resolution, with hash function H(K) = K mod 5.

```

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>3</td>
<td>22</td>
<td>13</td>
<td>7</td>
</tr>
</tbody>
</table>
```

a. [10 pts] Write down, in order of first insertion to last, the 5 keys that were inserted. (There may be more than one solution to this.)

Inserted keys: _____, _____, _____, _____, _____

b. [3 pts] Suppose all keys are equally likely. In this table, what is the exact average number of probes (comparisons) for a successful search? (Please show your work.)

_____________