READ THIS NOW!

- Print your name in the space provided below.
- There are 9 short-answer questions, priced as marked. The maximum score is 100.
- When you have finished, sign the pledge at the bottom of this page and turn in the test and your fact sheet.
- Aside from the allowed one-page fact sheet, this is a closed-book, closed-notes examination.
- No laptops, calculators, cell phones or other electronic devices may be used during this examination.
- Until solutions are posted, you may not discuss this examination with any student who has not taken it.
- Failure to adhere to any of these restrictions is an Honor Code violation.

Name (Last, First) Solution
printed

Pledge: On my honor, I have neither given nor received unauthorized aid on this examination.

_____
signed

1. [12 points] Determine the exact count complexity function T(N) of the body of the following function. Assume that calls to operator new take time N. You may assume any other single operations take time 1, as defined in the course notes. Your answer for T(N) should be in simplest form. Show supporting work!

```
public static void Q1(int[] A, int N) {     // Line 1

   int[] B = new int[N];                    // Line 2  1 + N

   for (int i = 0; i < N; i++) {            // Line 3  1 before, 2 per pass,
                                            //         1 to exit

      B[i] = 0;                             // Line 4  2

      for (int j = 1; j <= N; j = 2*j) {    // Line 5  1 before, 3 per pass,
                                            //         1 to exit

         B[i] = B[i] + i*j;                 // Line 6  5
      }

      A[N-1-i] = B[i];                      // Line 7  5
   }
}
```

$$T(N) = 1 + N + 1 + \sum_{i=0}^{N-1}\left(2 + 2 + 1 + \sum_{pass=1}^{1+\log N} (3+5) + 1 + 5 \right) + 1$$

$$= \sum_{i=0}^{N-1}\left(\sum_{pass=1}^{1+\log N} (8) + 11 \right) + N + 3$$

$$= \sum_{i=0}^{N-1}\left(8(1+\log N) + 11 \right) + N + 3$$

$$= \sum_{i=0}^{N-1}\left(8\log N + 19 \right) + N + 3$$

$$= N(8\log N + 19) + N + 3$$

$$= 8N\log N + 20N + 3$$

2. Assume that $f(n)$ and $g(n)$ are positive-valued functions defined on the set of non-negative integers.

a) [6 points] What does the fact given below imply regarding any big-O, big-Ω and/or big-Θ relationships between the functions? Be complete and precise. No justifications are needed.

$\forall n \geq 1000, f(n) \leq 7g(n)$

Straight from the definition, this implies that $f(n)$ **is** $O(g(n))$

Of course, this also implies that $g(n)$ **is** $\Omega(f(n))$**. But it does not imply any other relationships.**

b) [6 points] What does the fact given below imply regarding any big-O, big-Ω and/or big-Θ relationships between the functions? Be complete and precise. No justifications are needed.

$\forall n \geq 42, 3g(n) \leq f(n) \leq 5g(n)$

Straight from the definition, this implies that $f(n)$ **is** $\Theta(g(n))$

Of course, this also implies that $g(n)$ **is** $\Theta(f(n))$**, and every conceivable big-O and big-Ω relationship.**

3. [10 points] Consider the following functions:

$$f(n) = n^{1.01} + n\log n \qquad g(n) = n^{1.01} \qquad h(n) = n\log n$$

Which of the following is true? f is $\Theta(g)$. f is $\Theta(h)$.

Prove your conclusion is correct by using theorems covered in class. Note: the theorem that says the Θ-category is determined by the dominant term doesn't help since none of the theorems directly say which is the dominant term in this case.

$$\lim_{n\to\infty}\frac{f(n)}{g(n)} = \lim_{n\to\infty}\frac{n^{1.01}+n\log n}{n^{1.01}} = \lim_{n\to\infty}\left(\frac{n^{1.01}}{n^{1.01}}+\frac{n\log n}{n^{1.01}}\right)$$
$$= 1+\lim_{n\to\infty}\left(\frac{\log n}{n^{0.01}}\right) = 1+\lim_{n\to\infty}\left(\frac{1/n\ln 2}{0.01n^{-0.99}}\right)$$
$$= 1+\lim_{n\to\infty}\left(\frac{100}{n^{0.01}}\right) = 1$$

This implies that f is $\Theta(g)$.

If you tried it the other way around:

$$\lim_{n\to\infty}\frac{f(n)}{h(n)} = \lim_{n\to\infty}\frac{n^{1.01}+n\log n}{n\log n} = \lim_{n\to\infty}\left(\frac{n^{0.01}}{\log n}\right)+1$$
$$= \lim_{n\to\infty}\left(\frac{0.01n^{-0.99}}{1/n\ln 2}\right)+1 = \lim_{n\to\infty}\left(\frac{n^{0.01}}{100\ln 2}\right)+1 = \infty$$

This implies that f is strictly $\Omega(h)$, and so not $\Theta(h)$.

4. [12 points]

a) In the context of hashing, what is a *collision*?

A collision occurs when a hashing scheme maps two different key values into the same table slot.

b) Make the following assumptions about a particular hash table implementation:

- The table is reasonably sized, say twice the number of data objects that will be stored in it.
- The hash function H() is 1-1 from the set of actual key values into the set of integers.

Is it possible that collisions still occur when records are inserted to the table?

Yes.

If collisions could still occur, what <u>besides defects in the hash function</u> could cause a collision to occur.

Even if the table size is reasonable, when we mod the hash function value by the table size we may get the same table index, and hence a collision.

The basic issue here is that being 1-1 from the set of actual keys into the (infinite) set of integers is not the same as being 1-1 into the (very finite) set of valid table indices.

5. [10 points] A hash table implementation uses some form of probing (could be any probing strategy we discussed) to resolve collisions, but the implementer did not employ tombstones when performing deletion. Instead, the implementer simply reset a table slot to EMPTY if the element in that slot was deleted.

As a result, if a single element is deleted from the table, some subsequent searches may fail. Describe precisely, in a single sentence, which elements (that were already in the table when the deletion was performed) cannot now be found.

We will not be able to find any element that was inserted while the deleted element was in the table, and whose insertion required probing that hit (and so went beyond) the slot that contained the deleted element.

---

6. [10 points] Recall Pugh's probabilistic skip list. Assume that an implementation does use a random number generator that has the necessary property of producing a uniform distribution of values, and in fact that about 1/2 the nodes will be in level 0, 1/4 will be in level 1, 1/8 will be in level 2,and so forth.

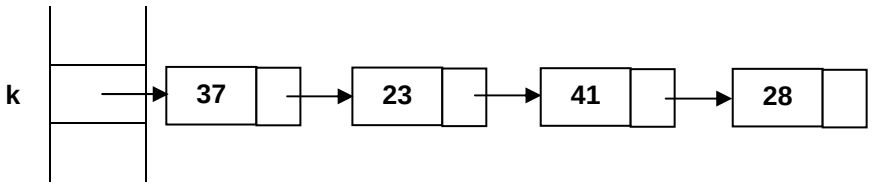
Explain how it is still possible that most of the searches in the skip list will require well over log N comparisons.

Even with the correct proportion of skip node types, it's possible that all the level-0 nodes fall in a contiguous sequence, and so do all the level-1 nodes, and so forth.

In that case, once a search reaches such a sequence of same-level nodes, it deteriorates to a simple linear traversal.

7. [12 points] Two developers are told to implement a hashing scheme for the same set of data records, and to use the same hash function and table size. One developer decides to use linear probing to resolve collisions. The other developer decides to use chaining with singly-linked lists in each table slot, appending new records to the lists.

When testing her implementation, the second developer discovers that one of the table slots ends up in the following state (displaying key values for the records):



Given that information, should the first developer be confident that if the record with key 28 is searched for in his implementation, exactly 3 probe steps will be required (accessing the home slot doesn't count as a probe step)? Explain clearly why or why not. If not, could the number of required probe steps be less than x? more than x?

When 28 was inserted to the table that used linear probing, the probing must have hit the slots that contained 37 and 23 and 41. So it's not possible that fewer probe steps will be required.

On the other hand, it's entirely possible that additional slots will also be hit during the linear probing, due to the tendency of linear probing to create contiguous clusters of filled slots; in that case more probe steps will be required.

---

8. [10 points] Recall the AVL tree. Considering only insertion operations, what is that the rotations do physically that causes the AVL to be superior to a plain BST? Illustrate your reasoning with an example!

Following an insertion, the effect of any rotations would be to cause the subtree where the insertion occurred to be one level shorter than it would have been without any rotations.

See the AVL notes for illustrative examples.

9. [12 points] Consider using a PR quadtree (with bucket size 1) to organize data objects that have integer coordinates in the square region of the xy-plane that is bounded between the points (0, 0) and (1024, 1024).

a) If there are 80 data points, what is the smallest number of levels the quadtree could have? Justify your conclusion.

Since the branching factor of a quadtree node is 4, the level 0 could contain 1 node, level 1 could contain up to 4 nodes, level 2 could contain up to 16 nodes, level 3 could contain up to 64 nodes, and level 4 could contain up to 256 nodes.

So, there would have to be at least 5 levels in order to have enough leaf nodes to store 80 data points.

b) Suppose the quadtree contains 50 data points, and that every leaf represents a region that is 64 x 64 or smaller. Suppose that another data point is inserted to the quadtree, and that it is exactly 2 units away from another data point that is already in the quadtree. Will the insertion of the new data point necessarily cause a leaf to split? If yes, explain why. If no, give an example illustrating why not.

A split will only occur if the new data point falls into an existing leaf node; that is, if it falls into a subregion that already contains a data point.

If the new point falls in an empty subregion, but is very close to a boundary and the adjacent subregion contains a data point that is also very close to the boundary, then those data points could be a distance 2 apart and there would be no split.