**Name:**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ SID **#:**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**True/False** (6 points each)

*Indicate whether the sentence or statement is true or false.*

\_\_\_\_1. f(n) = 4n + 2n2 + 5 ∈ O(n).

\_\_\_\_2. A sequential search of a list assumes that the list is in ascending order.

\_\_\_\_3. Binary search can be performed on both sorted and unsorted lists

\_\_\_\_4. The load factor is equal to the number of records in the hash table divided by the size of the hash table.

**Multiple Choice** (7 points each)

*Identify the letter of the choice that best completes the statement or answers the question.*

\_\_\_\_5. What is the number of key comparisons made when searching a list L of length n for an item using binary search?

|  |  |  |  |
| --- | --- | --- | --- |
| a. | O(log2 n) | c. | O(1) |
| b. | O(nlog2n) | d. | O(n) |

\_\_\_\_6. The merge sort algorithm uses the \_\_\_\_ technique to sort a list.

|  |  |  |  |
| --- | --- | --- | --- |
| a. | divide and conquer | c. | back tracking |
| b. | hash | d. | None of the above |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] |
| list | B | D | F | H | L | N | P | R | U |

\_\_\_\_7. Using binary search algorithm, how many key comparisons would have to be made on the list above to find the letter P?

|  |  |  |  |
| --- | --- | --- | --- |
| a. | 7 | c. | 6 |
| b. | 2 | d. | 5 |

**Short-Answer questions**

*Answer the following questions in the space provided.*

1. (9 points each) For each of the following program fragments compute the worst-case asymptotic time complexity (as a function of n). Where it says “loop body” you can assume that a constant number of lines of code are there. Briefly explain how you obtained your answer.
2. a = n + 2; O (n) = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. for ( i = 0; i < 3; i++ ) {

for ( j = 0; j < n; j++ )

loop body;

} O (n) = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. for ( i = 0; i < n \* n =; i++ )

loop body; O (n) = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. int sum = 0;

for ( i = 0; i < n; i++ ) {

if ( n % 2 == 0 )

sum++;

else

sum = sum + n;

} O (n) = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. for ( i = 0; i < n – 1; i++ ) {

for ( j = i + 1; j <= n - 1; j++ )

loop body;

} O (n) = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. (10 points) Suppose that the keys 23, 66, 47, 87, 126, 140, 145, 467 and 99 are to be inserted in this order into the initial empty hash table given below. Use the modular arithmetic method of hashing, h(x)=x % HTSize, and quadratic probing to resolve collisions. Show the hash table after storing the keys and then compute the load factor α.

Hash Table

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

Load factor α = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_