Searching

Standard difficulty · 20 questions · 40 marks · Time: 41 minutes

Name: _____ Date: _____ Score: ____ / 40

Show your working for Sections B and C. Answers are on the last page — try every question first!

Section A — Choose the correct answer — 8 × 1 = 8 marks

1. Linear search compares the key with:

(a) only the last item **(b)** the item at key % size

(c) the middle item first **(d)** each item in turn from the start

2. Linear search looks for 64 in 78, 81, 73, 55, 75, 72, 95, 64. How many comparisons does it make?

(a) 1 **(b)** 7 **(c)** 8 **(d)** 9

3. Linear search looks for 42 in 38, 14, 59, 3, 89, 64, 88. How many comparisons does it make before reporting that 42 is not present?

(a) 7 **(b)** 8 **(c)** 0 **(d)** 6

4. Binary search looks for 67 in the sorted list 11, 13, 38, 39, 45, 47, 50, 67, 70, 73, 83, 94, 97. How many comparisons with a middle item does it make?

(a) 3 **(b)** 8 **(c)** 2 **(d)** 4

5. In binary search on the sorted list 13, 48, 49, 53, 56, 61, 76, 92 (indices from 0), which item is compared first with the key 92?

(a) 92 **(b)** 49 **(c)** 56 **(d)** 53

6. A hash table of size 10 uses h(key) = key % 10. At which index is 75 stored?

(a) 4 **(b)** 6 **(c)** 5 **(d)** 7

7. What is the largest number of comparisons binary search can need to find a key in a sorted list of 500 items?

(a) 250 **(b)** 10 **(c)** 8 **(d)** 9

8. What does this print?

```
L = [17, 20, 31, 60, 62, 63, 71, 82, 88, 91, 93]
key = 71
first, last = 0, len(L) - 1
count = 0
pos = None
while first <= last:
    mid = (first + last) // 2
    count = count + 1
    if L[mid] == key:
        pos = mid
        break
    elif L[mid] > key:
        last = mid - 1
    else:
        first = mid + 1
print(pos, count)
```

(a) 6 2 **(b)** 6 3 **(c)** 7 4 **(d)** 6 4

Section B — Short answer — 8 × 2 = 16 marks

9. What is searching?

10. Binary search looks for 60 in 3, 6, 21, 27, 35, 36, 48, 60, 64, 68, 78, 85 (indices from 0). Write the mid indices it looks at, in order, separated by commas.

11. The textbook's linearSearch(list, key) returns the position of key counting from 1. What does it return for key 14 in 65, 57, 73, 86, 6, 39, 42, 60, 14, 58, 90?

12. Keys 21, 10, 67, 84 are placed in a hash table of size 10 with h(key) = key % 10. (a) At which index does 21 go? (b) What is h(21)?

13. What does this print?

```
L = [7, 17, 21, 27, 33, 36, 47, 71, 81, 83, 87, 98]
key = 87
first, last = 0, len(L) - 1
count = 0
pos = None
while first <= last:
    mid = (first + last) // 2
    count = count + 1
    if L[mid] == key:
        pos = mid
        break
    elif L[mid] > key:
        last = mid - 1
    else:
        first = mid + 1
print(pos, count)
```

14. For key 94 in the sorted list 12, 18, 19, 22, 30, 39, 55, 62, 66, 71, 77, 78, 87, 90, 94, how many comparisons does (a) linear search make (b) binary search make?

15. Keys 49, 74, 62, 59, 29, 76 are inserted in this order into an empty hash table of 7 slots with h(key) = key % 7; a key whose slot is taken is not stored. Write the table as a Python list (None for an empty slot).

16. Keys 23, 59, 19, 16 go into a hash table of size 10 with h(key) = key % 10. Is there a collision? Answer Yes or No.

Section C — Word problems — 4 × 4 = 16 marks

17. Write a Python function for linear search that returns the position of the key (counting from 1), or None. Explain its best and worst cases.

18. L = 5, 17, 21, 23, 40, 43, 47, 59, 61, 62, 65, 67, 75, 81, 89 (indices from 0). Using binary search, (a) write the mid indices looked at when searching for 62; (b) do the same for 69; (c) how many comparisons would linear search make for each?

19. Keys 89, 53, 68, 86, 39 are inserted into an empty hash table of size 7 with h(key) = key % 7 (a key whose slot is taken is not stored). (a) Write the index of each key. (b) How many comparisons does searching for 89 need? (c) Searching for 80: which slot is checked, and what is found there?

20. (a) What does this program print? (b) Why must L be sorted? (c) What is the largest number of passes this loop can need for a list of this length?

```
L = [6, 7, 8, 24, 26, 27, 59, 63, 71, 84, 98]
key = 19
first, last = 0, len(L) - 1
count = 0
pos = None
while first <= last:
    mid = (first + last) // 2
    count = count + 1
    if L[mid] == key:
        pos = mid
        break
    elif L[mid] > key:
        last = mid - 1
    else:
        first = mid + 1
print(pos, count)
```

Want another sheet on this chapter? In the Ambiful Worksheets app every sheet is new, with its answer key, and your child never gets the same question twice.
Scan the code to get the free app, or open **worksheets.ambiful.com/get**

Answer key

Searching · AW-12CS-SRC-2026-V1 · For parents — fold back or tear off before handing the sheet to your child.

Q	Answer	Working
1	**(d) each item in turn from the start**	—
2	**(c) 8**	64 is item 8 from the left
3	**(a) 7**	a missing key is compared with every item
4	**(a) 3**	mid indices (from 0): 6, 9, 7
5	**(d) 53**	mid = (0 + 7) // 2 = 3; then mids 3, 5, 6, 7
6	**(c) 5**	75 = 10 × 7 + 5
7	**(d) 9**	the list halves each time: 9 halvings reduce 500 items to 1
8	**(b) 6 3**	mid = (first + last) // 2; the search halves the range after each comparison
9	**Finding whether a key is present in a collection of items and, if so, where.**	—
10	**5, 8, 6, 7**	found at index 7
11	**9**	9 comparisons
12	**(a) 1; (b) 1**	remainders: 21 → 1, 10 → 0, 67 → 7, 84 → 4
13	**10 3**	pos stays None when the key is not found; count is the number of passes
14	**(a) 15; (b) 4**	binary search mids: 7, 11, 13, 14
15	**[49, 29, None, 59, 74, None, 62]**	49 → 0, 74 → 4, 62 → 6, 59 → 3, 29 → 1, 76 → 6; 76 clashes
16	**Yes**	remainders: 3, 9, 9, 6
17	**def linearSearch(L, key): for i in range(len(L)): if L[i] == key: return i + 1; return None. Best case: the key is first, 1 comparison; worst case: the key is last or absent, n comparisons.**	—
18	**(a) 7, 11, 9; (b) 7, 11, 13, 12, not found; (c) 10 and 15**	mid = (first + last) // 2; move first or last past mid each time
19	**(a) 89 → 5, 53 → 4, 68 → 5, 86 → 2, 39 → 4; (b) 1; (c) slot 3, which holds nothing, so 80 is not present**	hashing finds a key with one calculation and one comparison
20	**(a) None 3; (b) binary search throws away the half that cannot hold the key, which is only correct when the items are in order; (c) 4**	each pass halves the range

Ambiful Worksheets · worksheets.ambiful.com · Not affiliated with CBSE / NCERT