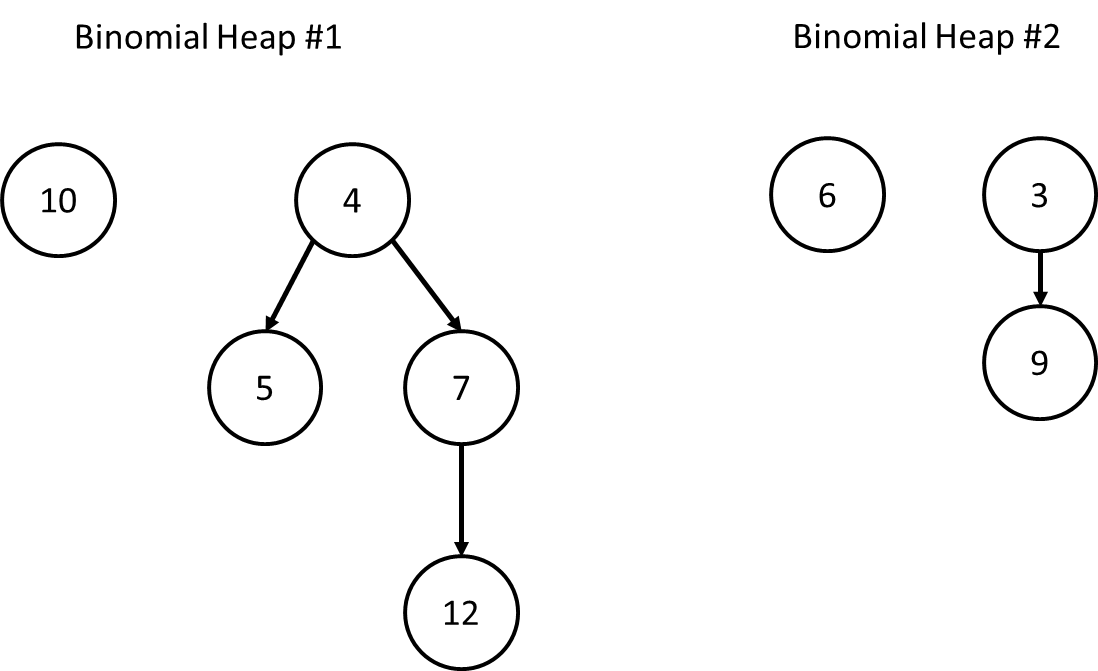
4. **Binary Heaps** Starting with an empty binary **min heap**, show the following.

1. [3] The final state of the heap, in tree form, after adding in the values: 5, 4, 3, 6, 7, 8, 10, 2, 9, 1
2. [2] The state of the heap, in tree form, after two Dequeue() operations
3. [1] The final, array-based version of the heap

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

5. [3] Merge the following two **binomial heaps**



7. [3] The item "B" hashes to array index 3. Insert B into the following **hopscotch** hashtable whose max distance is 4.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | C | E | G | D | H | I |  |  |  |
| 1100 | 0100 | 0010 | 1000 | 0000 | 1100 | 0000 |  |  |  |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

RESULT:

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

**20. [2]** Given the quadratic probe(i) = (i^2 + 1) % 10, insert the value H, which hashes to array location 1.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F |  |  |  |  |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

**19. [2] Cuckoo Hashtables.** Given the following hash results, add the value **"A"** into the **first array** of the cuckoo hash below:

Hashing results:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A: 2,5 | B: 1,5 | C: 2,7 | D: 9,3 | E: 1,2 | F: 7,2 | G: 3,5 | I: 3,3 | J: 0,5 | K: 2,3 |

Array 1:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| J | B | K | I |  |  |  |  |  |  |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Array 2:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | E | D |  | G |  | C |  |  |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

**YOUR ANSWER: After inserting 'A'**

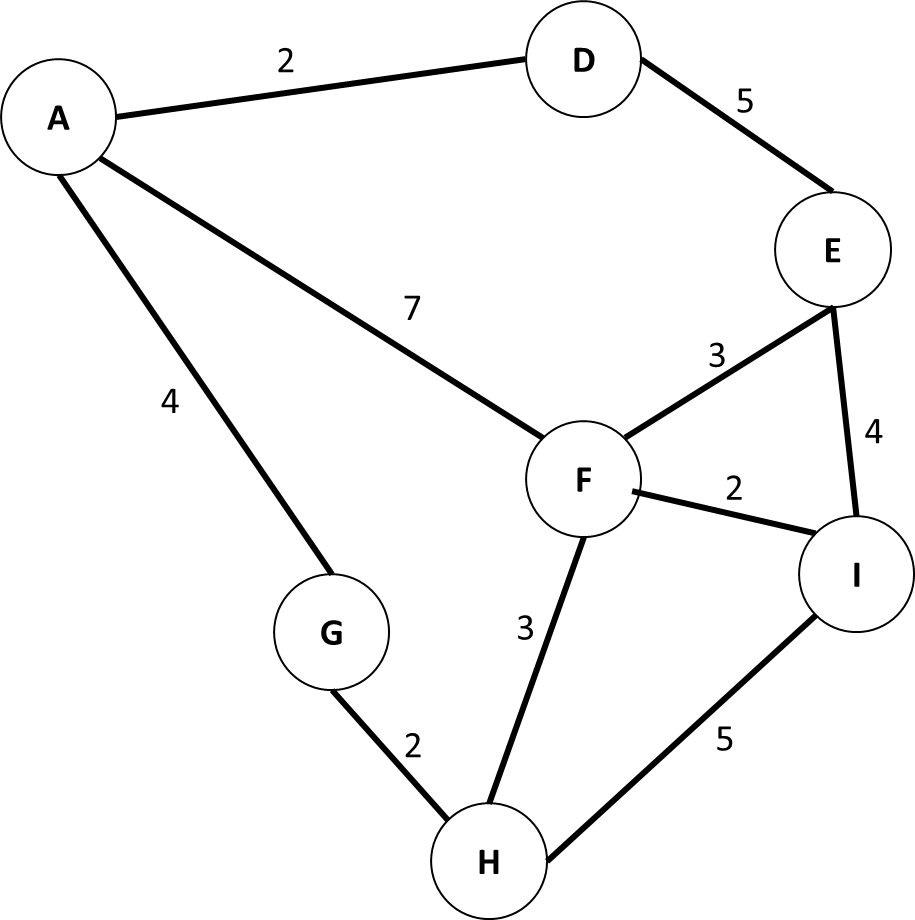
Array 1:

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

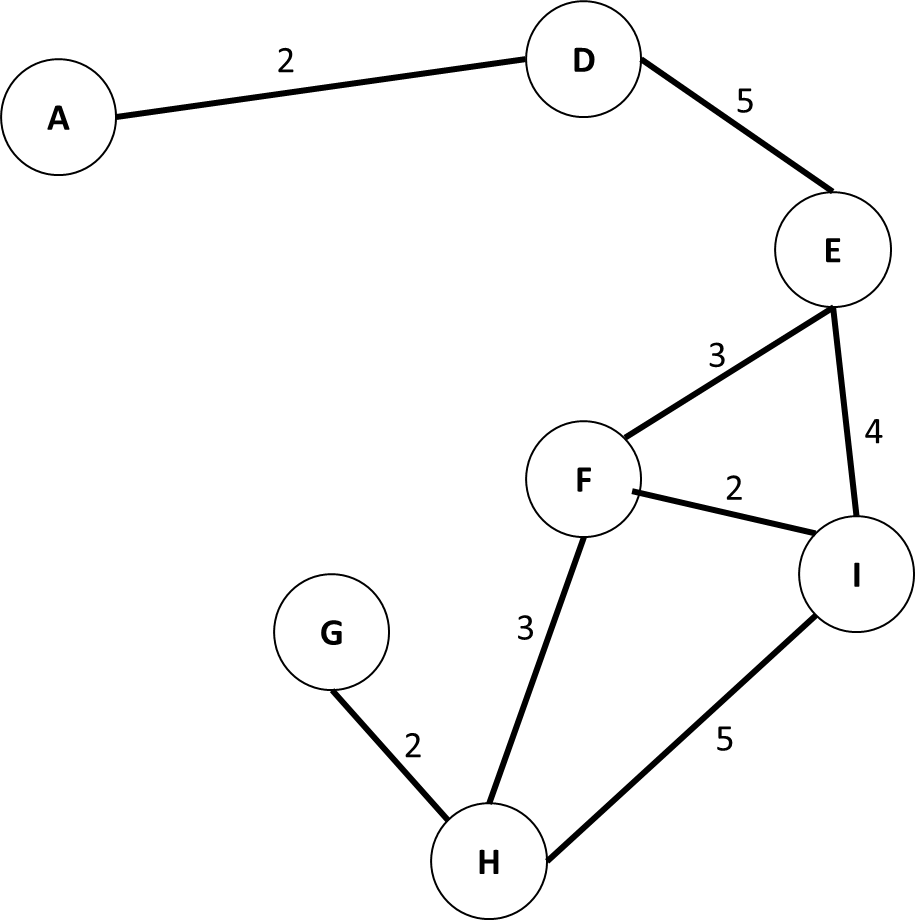
Array 2:

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

Construct an MST for the following graph:



Build a DFS articulation tree for the following graph:



Show the result of quicksort after one iteration on the following array:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | 9 | 1 | 12 | 14 | 8 | 1 | 7 | 2 | 6 |

Perform radix sort on the above array