

Questions

1. (10 points) For each of the following program fragments, give an analysis of the running time.

- | | |
|---|---|
| <p>(1) <code>sum = 0;</code>
 <code>for (i = 0; i < N; i++)</code>
 <code> for (j = 0; j < i; j++)</code>
 <code> sum ++;</code></p> | <p>(3) <code>sum = 0;</code>
 <code>for (i = 0; i < n; i++)</code>
 <code> for (j = 0; j < n; j++)</code>
 <code> sum++;</code></p> |
| <p>(2) <code>sum = 0;</code>
 <code>for (i = 1; i < N; i *= 2)</code>
 <code> sum ++;</code></p> | <p>(4) <code>sum = 0;</code>
 <code>for (i = 1; i < n; i++)</code>
 <code> for (j = 1; j < i * i; j++)</code>
 <code> if (j % i == 0)</code>
 <code> sum++;</code></p> |
| <p>(5) <code>sum = 0;</code>
 <code>for (i = 1; i < n; i *= 2)</code>
 <code> for (j = 0; j < n; j += i)</code>
 <code> sum++;</code></p> | |

2. (12 points) Given input 4371, 1323, 6173, 4199, 4344, 9679, 1989, a hash table with size 10 and a hash function $h(x) = x \% 10$, show the resulting

- Separate chaining hash table
- Open addressing hash table using linear probing

3. (12 points)

- Write a function to reverse a singly link list. Your function will return a new list with new elements (You are not allowed to modify original link list). Prototype: **Listptr reverse_list(Listptr l)**
- Write the function `palindrome` which returns 1 if the list contains the same elements in the same order as its reversed list, 0 otherwise. You will use the function above. Prototype: **int palindrome(Listptr l)**

4. (12 points)

- Write the function **int number_of_leaves(Treenodeptr r)** which returns the number of leaf nodes in the tree with root r .
- Write the function **void print_decreasing_order(Treenodeptr r)** which prints the elements in the binary search tree with root r in decreasing order.

5. (12 points) Sort the integer sequence 3, 1, 4, 1, 5, 9, 2, 6 using

- insertion sort
- merge sort

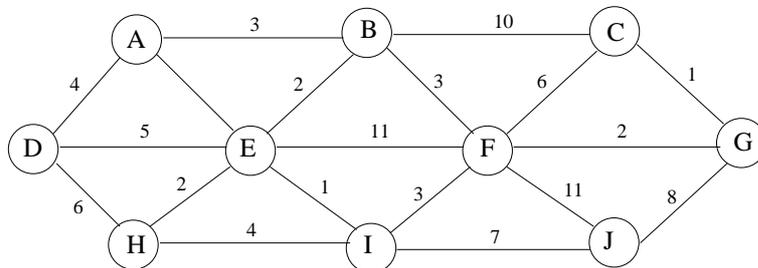
6. (8 points) Show the result of each operation in the following program.

```
void main(){
    Stackptr s;
    s = empty_stack();
    push(s, 4);
    push(s, 1);
    push(s, 3);
    x = pop(s);
    push(s, 8);
    y = pop(s);
    z = pop(s);
}
```

7. (12 points)

- Implement `insert_back` for doubly link lists. Prototype: `void insert_back(Doublylistptr l, int data)`
- Implement `remove_back` for doubly link lists. Prototype: `int remove_back(Doublylistptr l)`

8. (10 points) Find the minimum spanning tree of the following graph using either Prim or Kruskal's algorithm.



9. (12 points) Write a function which reads a graph from a text file and returns the graph in the adjacencylist representation. First line will contain the number of nodes and the number of edges in the graph. Each of the following lines (2, 3, 4, ...) contains two numbers representing the fromNode (the node where edge starts) and toNode (the node where edge ends) of the edge. Prototype: `Graphptr read_graph(char* filename)`

10. (10 points) You are given a list of league game scores (there are no ties). If all teams have at least one win and a loss, we can generally prove that any team is better than any other. For instance, in the 3-team league we have the following results Fenerbahce beats Galatasaray, Galatasaray beats Besiktas, Besiktas beats Fenerbahce. Then we can prove that Fenerbahce is better than Besiktas, because Fenerbahce beats Galatasaray, which in turn beats Besiktas. Given a list of game scores and two teams X and Y, find if X beats Y.