CS202 HW 4

Elifsena Öz

22002245

Section 1

Question 3:

Worst-case Asymptotic Complexity of Insertion:

The graph is implemented as an adjacency list. Therefore, to add a flight to a specific airport from the array of airports, we can access the list of flights in O(1) time. Since the list is sorted, at worst case scenario we insert to the end of the list which means that we trace N – 1 elements in the list, O(N – 1) time. In total insertion has time complexity of O(1) + O(N – 1) = O (N).

Worst-case Asymptotic Complexity of List:

To list the flights from a specific airport, we can access the linked list for the airport in O(1) time. At worst case there can be N – 1 flights from one airport to another. Therefore, we will have to access N – 1 flights from the list, which has complexity of O(N – 1). In total List operation has time complexity of O(1) + O(N – 1) = O (N).

Worst-case Asymptotic Complexity of Shortest Path:

To find the shortest path from one airport to another, we use Dijkstra's Shortest Path Algorithm. First we determine the weight of each airport from the beginning in O(N) time. Then we visit each airport one by one choosing the smallest weighted unvisited airport and look at the duration of each flight from that airport. Finding the minimum weighted unvisited airport takes O(N – 1) time. Accessing each flight from the minimum weighted unvisited airport takes O(N – 1) time at worst case scenario (there are connections to all other airports). To print the path we also look at elements from the array which stores the previous airport for each airport, this operation takes O(N) time. All together the complexity is $O(N) + O((N - 1)^2) + O(N) = O(N^2)$.

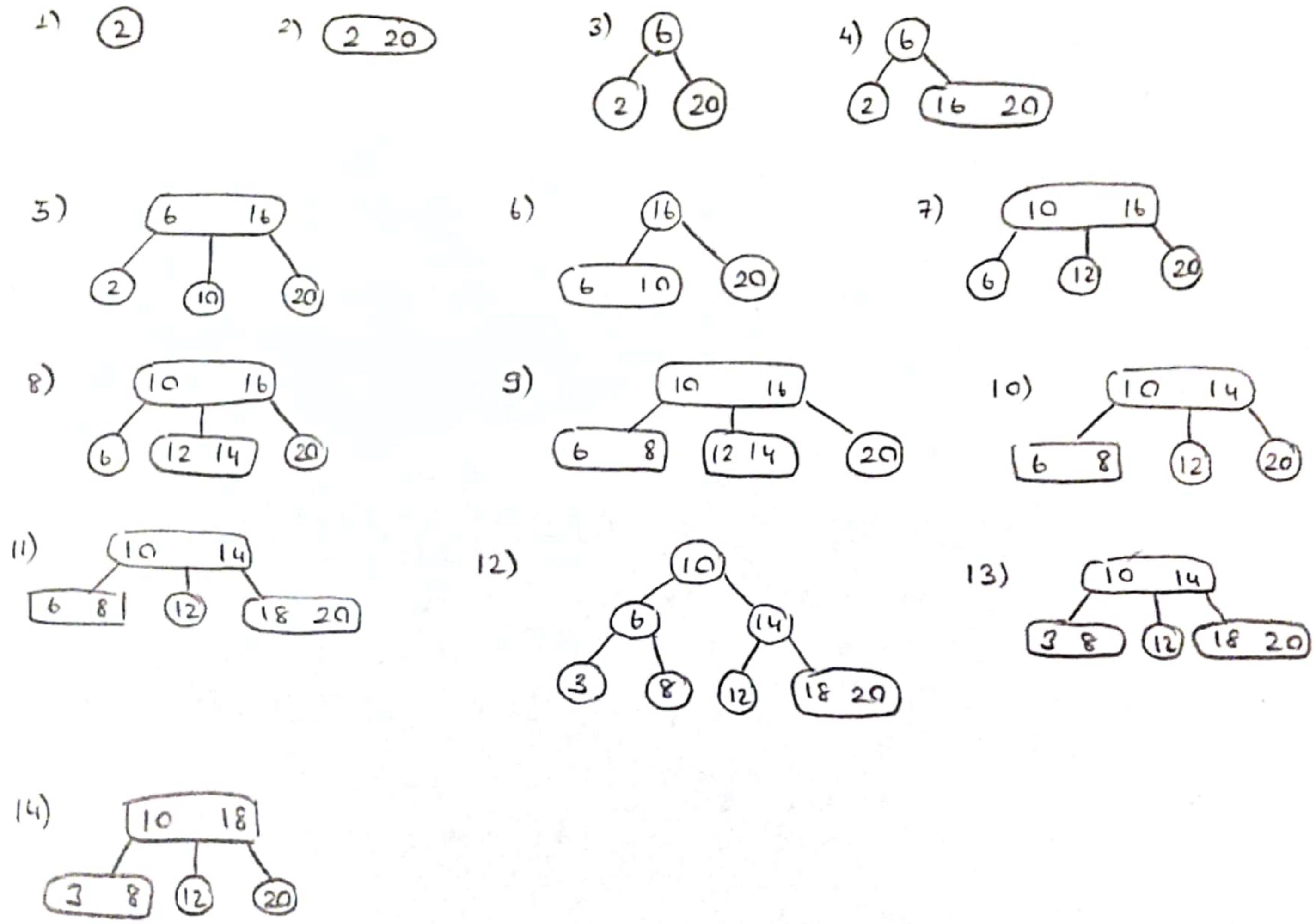
Worst-case Asymptotic Complexity of Minimize Costs:

To minimize cost, we use Prim's Algorithm. This algorithm looks at the shortest path from one visited airport to an unvisited airport which takes $O(N^2)$ in worst case scenario. The algorithm repeats this step for each airport until all airports are visited. This operation in total takes $O(N^3)$ time. Before and after we use this algorithm we calculate the cost of the flights in total by looking at each flight from each airport. This operation takes $O(N^2)$ time at worst case. In total minimizing the costs takes $O(N^3) + O(2 * N^2) = O(N^3)$.

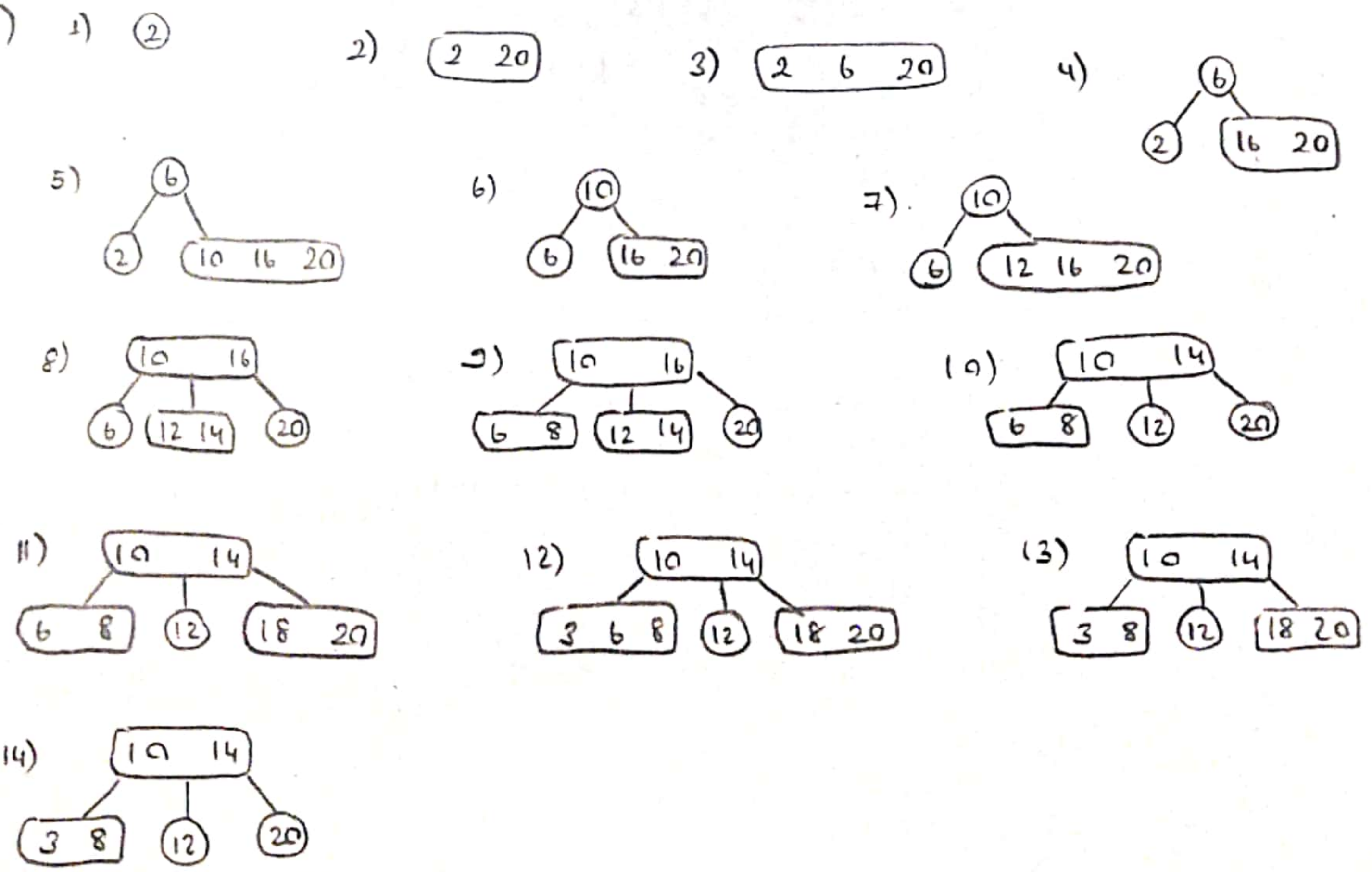
Question 1

Elifsena Öz
22002245

a)



b)



Question 2

Elifsena Öz
22002245

a)

0	26
1	
2	54
3	
4	17
5	69
6	45
7	58
8	32
9	60
10	
11	
12	64

45 mod 13 → 6
64 mod 13 → 12
54 mod 13 → 2
17 mod 13 → 4
69 mod 13 → 4 → 4 + 1 = 5
58 mod 13 → 6 → 6 + 1 = 7
32 mod 13 → 6 → 6 + 1 = 7 → 7 + 1 = 8
60 mod 13 → 8 → 8 + 1 = 9
26 mod 13 → 0

Try 54, 17, 45, 69, 60, 64, 32, 58, 26
Successful Search
av number of probes = $\frac{(1+1+1+2+2+1+3+2+1)}{9} = \frac{14}{9}$

Try 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
Unsuccessful Search
av number of probes = $\frac{(1+2+1+6+5+4+3+2+1+1+3+7+2)}{13} = \frac{38}{13}$

b)

0	26
1	
2	54
3	
4	17
5	69
6	45
7	58
8	60
9	
10	32
11	
12	64

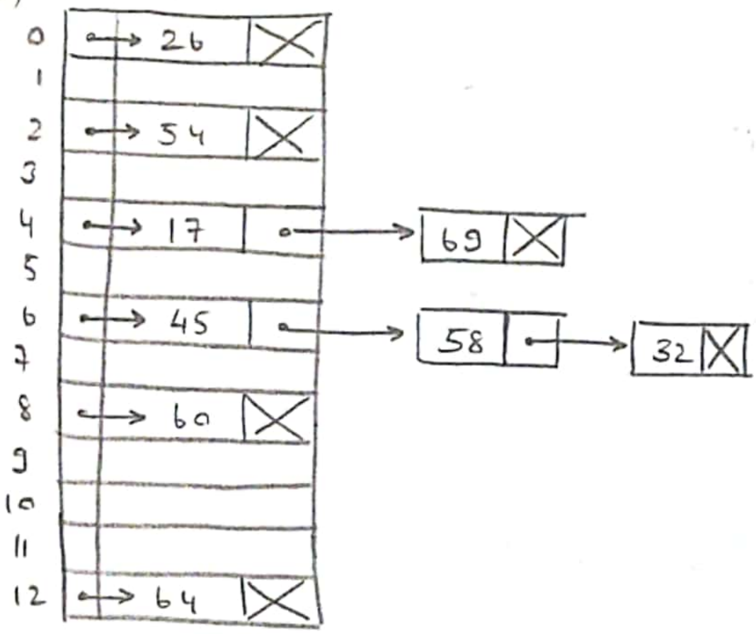
45 mod 13 → 6
64 mod 13 → 12
54 mod 13 → 2
17 mod 13 → 4
69 mod 13 → 4 → $4 + 1^2 = 5$
58 mod 13 → 6 → $6 + 1^2 = 7$
32 mod 13 → 6 → $6 + 1^2 = 7$ → $6 + 2^2 = 10$
60 mod 13 → 8
26 mod 13 → 0

Try 54, 17, 45, 69, 60, 64, 32, 58, 26
Successful Search
av number of probes = $\frac{(1+1+1+2+1+1+3+2+1)}{9} = \frac{13}{9}$

Try 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
Unsuccessful Search
av number of probes = $\frac{1+1+1+1+2+2+2+2+3+3+3+5+7}{13} = \frac{33}{13}$

c)

Elifsena Öz
22002245



Using the same test cases from part a & b

Successful Search

$$av = \frac{1+1+1+1+1+1+1+2+2+3}{9} = \frac{13}{9}$$

Unsuccesful Search

$$av = \frac{1+1+1+1+1+1+1+1+2+2+2+2+3+4}{13} = \frac{22}{13}$$