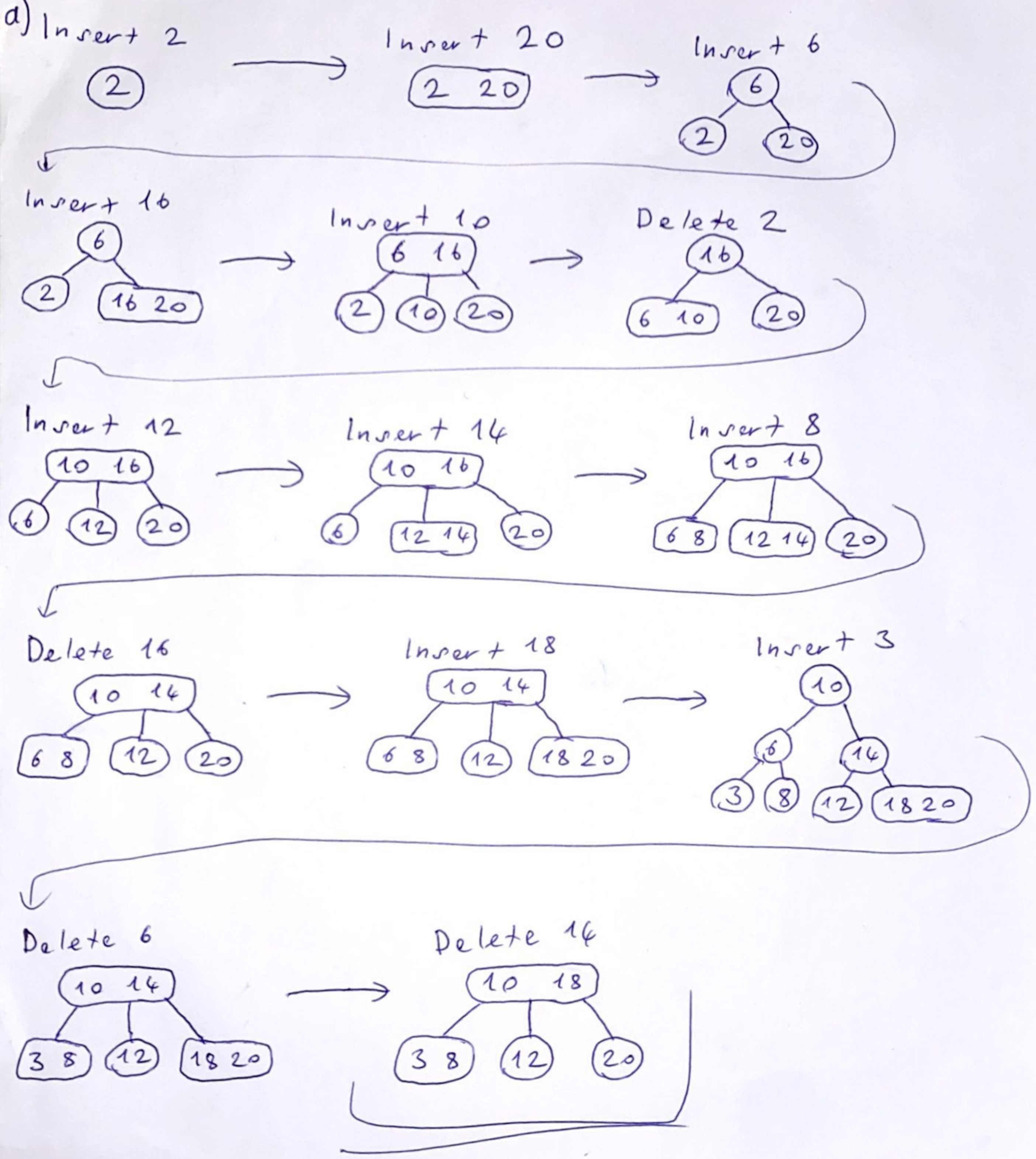
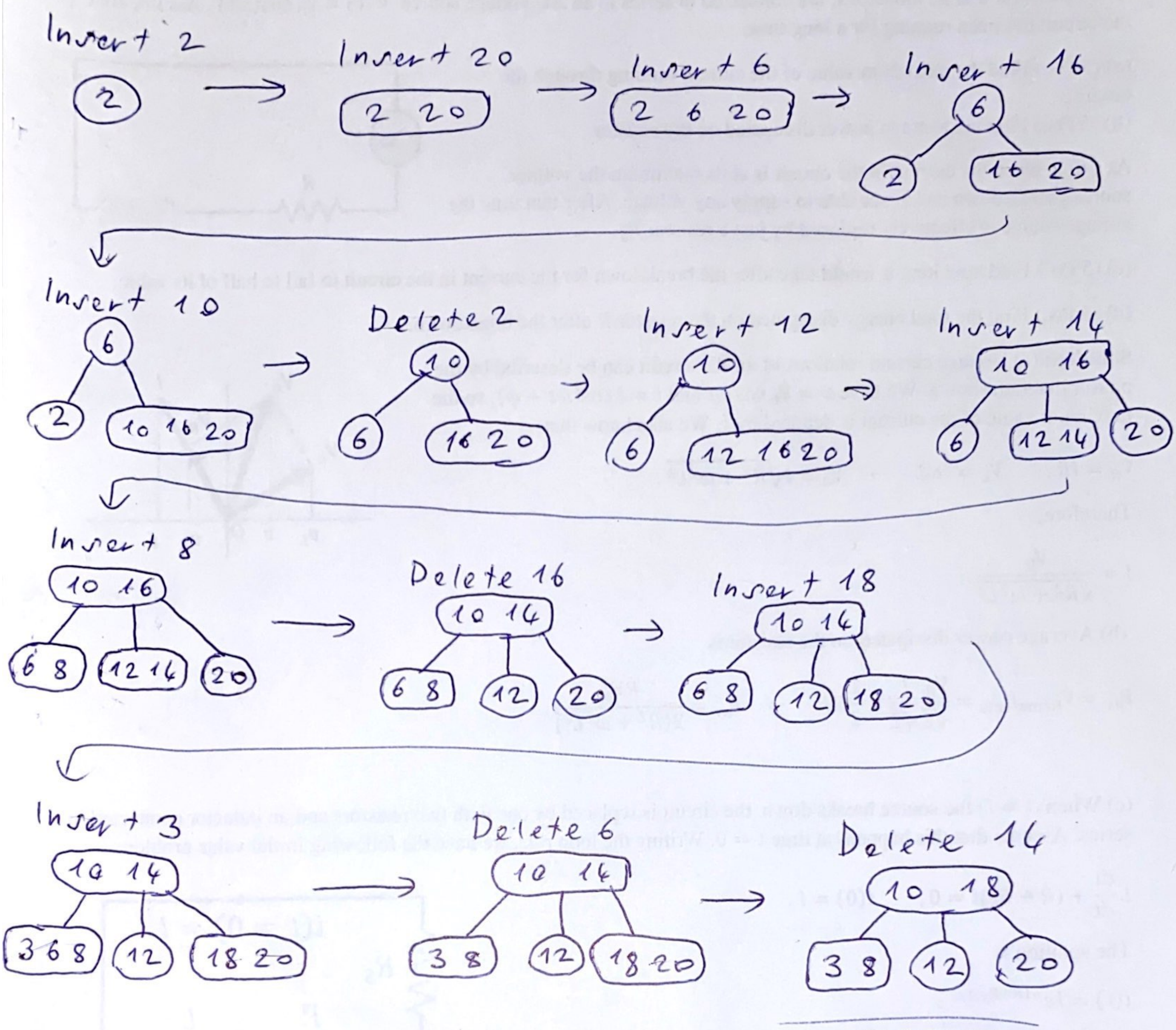
Mehmet Onur Uysal
22002809
CS202 section 1
HW4

Question 1

a)



b)


Insert 2
Insert 20
Insert 6
Insert 16
Insert 10
Delete 2
Insert 12
Insert 14
Insert 8
Delete 16
Insert 18
Insert 3
Delete 6
Delete 14

Question 2

a)

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

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 → 6+2 = 8

60 mod 13 = 8 → 8+1 = 9

26 mod 13 = 0

Average number of probes for successful search $= \frac{1}{2}\left(1 + \frac{1}{1-\frac{9}{13}}\right) = 2.125$

Unsuccessful search $= \frac{1}{2}\left(1 + \frac{1}{1-\left(\frac{9}{13}\right)^2}\right) = 5.78$

b)

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

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 → $6+1^2$ → $6+2^2 = 10$

60 mod 13 = 8

26 mod 13 = 0

Average number of probes for successful search $= \frac{-\ln\left(1-\frac{9}{13}\right)}{9/13} = \underline{1.7}$

unsuccessful search $= \frac{1}{1-\frac{9}{13}} = \underline{3.25}$

c)

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

Average number of probes for

successful search $= 1 + \frac{9/13}{2} = \underline{1.34}$

unsuccesful search $= \frac{9}{13} = \underline{0.69}$

Scanned with CamScanner

Question 3

For this part, I implemented a weighted undirected graph. In the graph, I have an array of linked lists every one of which stores the airport number and the duration of the flight from the corresponding airport.

Insert Operation

I add new nodes at the start of each list so it takes O(1) time to insert. I do the insertion twice (for each direction) and show information about new flight. There is no worst case for the operation.

List operation

List operation grows linearly depending on the number of flights from the airport. Algorithm goes through all nodes in the list so it takes O(N) time. Worst case is when all the flights are from that specific airport. Then we would have iterations as the number of all flights.

Shortest path operation

I created parallel arrays to store the distance, visited vertices and resulting strings. Time complexity of the algorithm is $O(N^2)$ because we check for new paths for each unvisited vertex.

Minimize costs operation

For this operation I used a min heap to store the flights starting from the first. To find the minimum spanning tree, I added the flight with least duration to the new graph. Heap operations take O(log N) time and since I do it for all vertices worst case time complexity is O(NlogN).