1.) a.) $T(n) = 2T(n/2) + C$

1				C
			C	C
		$C\ C$		$C\ C$

1	n	C
2	$n/2$	$2C$
3	$n/2^2$	2^2C
k	$n/2^k$	2^kC

$n/2^{k-1} = 2 \qquad k = \log n$

$n = 2^k$

$\rightarrow C(2^0 + 2^1 + 2^2 \dots + 2^{k-1})$

$C(2^k - 1)$

$C(2^{\log n} - 1)$

$C(n-1)$.

$Cn - C$

$\boxed{\in O(n)}$

---

b.) $T(n) = 2T(n/4) + Cn$

Cn

$C\frac{n}{4} \quad C\frac{n}{4}$

$C\frac{n}{16}\ C\frac{n}{16} \quad C\frac{n}{16}\ C\frac{n}{16}$

$8 \cdot \frac{n}{64}$

1	n	Cn
2	$n/4$	$C\frac{n}{2}$
3	$n/4^2$	$C\frac{n}{4}$
4	$n/4^3$	$C\frac{n}{8}$
k	$n/4^{k-1}$	$C\frac{n}{2^{k-1}}$

$n/4^{k-1} = 2$

$2n = 4^k$

$\log_4(2n) = k$

$\rightarrow Cn\left(\frac{1}{2^0} + \frac{1}{2^1} + \frac{1}{2^2} \dots \frac{1}{2^{k-1}}\right)$

$Cn\left(\left(\frac{1}{2}\right)^0 + \left(\frac{1}{2}\right)^1 + \left(\frac{1}{2}\right)^2 \dots \left(\frac{1}{2}\right)^{k-1}\right)$

$$\frac{Cn\left(\left(\frac{1}{2}\right)^k - 1\right)}{-\frac{1}{2}}$$

$2Cn\left(1 - \left(\frac{1}{2}\right)^k\right)$

$2Cn\left(1 - \left(\frac{1}{2}\right)^{\log_4(2n)}\right)$

$\leq 2Cn$

$\boxed{\in O(n)}$

2.) Find_Majority(S): S is set of bank cards

```
if |S| is 1                                   | Base case
        return (true, S[0])

Divide S in half to C1, C2
(res1, B1) = Find_Majority(C1)                | res is true if
(res2, B2) = Find_Majority(C2)                |   B is majority
                                              |   2T(n/2)

if(res1)
    num1 = count equivalences w/ B1 in S      | O(n)
    if(num1 > |S|/2) return (true, B1)

if(res2)
    num2 = count eq w/ B2 in S                | O(n)
    if(num2 > |S|/2) return (true, B2)

return (false, {})                            | No majority so no card
```

Only possible for a card to be majority in S if it is a majority in one half of the cards, C_1 or C_2.

Break down until 1 card which has to be majority in S of 1 card.

Subsequent/higher levels check if majorities of halves are still majorities of larger S.

$T(n) = 2T(n/2) + Cn$

lvl	size	time
1	n	Cn
2	$n/2$	Cn
$k+1$	$n/2^k$	Cn

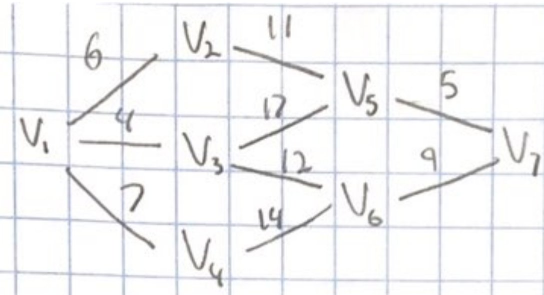
$\rightarrow (k+1)Cn$

$n/2^k = 1 \rightarrow k = \log n$

$\rightarrow Cn\log n + Cn$

$\rightarrow O(n\log n)$

3.)

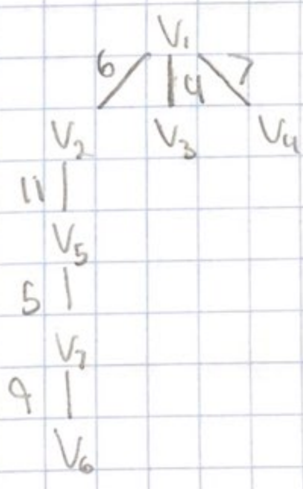


a.) Dijkstra's

Iter	V_1	V_2	V_3	V_4	V_5	V_6	V_7
0	0	∞	∞	∞	∞	∞	∞
1	0	6	4	7	∞	∞	∞
2	0	6	4	7	21	16	∞
3	0	6	4	7	17	16	∞
4	0	6	4	7	17	16	∞
5	0	6	4	7	17	16	25
6	0	6	4	7	17	16	22
7	0	6	4	7	17	16	22

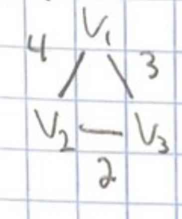
b.) Prim's

Iter	V_1	V_2	V_3	V_4	V_5	V_6	V_7
0	0	∞	∞	∞	∞	∞	∞
1	0	6^{V_1}	4^{V_1}	7^{V_1}			
2	0	6^{V_1}	4^{V_1}	7^{V_1}	17^{V_3}	12^{V_3}	
3	0	6^{V_1}	4^{V_1}	7^{V_1}	17^{V_3}	12^{V_3}	
4	0	6^{V_1}	4^{V_1}	7^{V_1}	11^{V_2}	12^{V_3}	
5	0	6^{V_1}	4^{V_1}	7^{V_1}	11^{V_2}	12^{V_3}	5^{V_5}
6	0	6^{V_1}	4^{V_1}	7^{V_1}	11^{V_2}	9^{V_7}	5^{V_5}
7	0	6^{V_1}	4^{V_1}	7^{V_1}	11^{V_2}	9^{V_7}	5^{V_5}

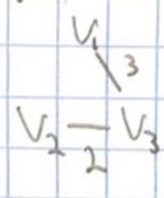


4.)

Graph



Only MST using Prim's



Shortest Paths from V_1

V_2 – 4

V_3 – 3

a.) Disprove by counterexample

Apply Prim's on V_1 in the graph above gives us the MST $V_1 \overset{3}{-} V_3 \overset{2}{-} V_2$

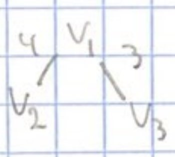
This does not contain the shortest path from $V_1 - V_2$ in the original graph.

In the original, V_1 and V_2 had a weight of 4 apart in the only MST using Prim's the weight is 5.

b.) Disprove by Counterexample

If we apply Dijkstra's on V_1 in the graph above we get the following table & graph

	V_1	V_2	V_3
1	0	4^{V_1}	3^{V_1}
2	0	4^{V_1}	3^{V_1}
3	0	4^{V_1}	3^{V_1}



The graph doesn't match the only possible MST for the original graph.