Q1.

Recurrence is $T(n) \leq T(\sqrt{n}) + T(n-\sqrt{n}) + cn$

$$T(n) \Rightarrow \begin{array}{c} cn \\ \swarrow \quad \searrow \\ T(n-\sqrt{n}) \quad T(\sqrt{n}) \end{array} \Rightarrow \begin{array}{c} cn \\ \swarrow \qquad\qquad \searrow \\ c(n-\sqrt{n}) \qquad\qquad c\sqrt{n} \\ \swarrow \quad \searrow \qquad\qquad \swarrow \quad \searrow \\ T(n-\sqrt{n}-\sqrt{n-\sqrt{n}}) \quad T(\sqrt{n-\sqrt{n}}) \quad T(\sqrt{n}-n^{1/4}) \quad T(n^{1/4}) \end{array}$$

We keep opening like this. Sum in each row is cn.

Let $H(n)$ denote height of tree, then the time complexity will be $cn \cdot H(n)$.

Observe that

$$H(n) = 1 + H(n-\sqrt{n}).$$

we can verify that $H(n) \leq 2\sqrt{n}$ as below.

Assume $H(k) \leq 2\sqrt{k} \quad \forall \; k \leq n-1.$

Then,

$$H^2(n) = \left(1 + H(n-\sqrt{n})\right)^2$$

$$\leq \left(1 + 2\sqrt{n-\sqrt{n}}\right)^2$$

$$= 1 + 4(n-\sqrt{n}) + 4\sqrt{n-\sqrt{n}}$$

$$= 4n + 4\left(\sqrt{n-\sqrt{n}} - \sqrt{n} + \frac{1}{4}\right)$$

$$\leq 4n$$

This shows $H(n) \leq 2\sqrt{n}$,
and thus time-complexity is $O(n\sqrt{n})$.

Q2.**

(a) We have n intervals $I_1 \ldots I_n$.

Task is :
Find a set $S \subseteq \{I_1 \ldots I_n\}$ such that $\forall j \leq n$
I_j overlaps with an interval in S,
and
$|S|$ is minimized.

NOTE : Intervals in S need not be Disjoint !

(b)

COMPUTE-OPT$(J := (I_1 \ldots I_n))$

① Let $I_\alpha = [s_\alpha, t_\alpha]$ be interval with least finish time.

② $Z = \text{overlap}(I_\alpha)$

③ Let $I_\beta = [s_\beta, t_\beta]$ be interval in Z with largest finish time.

④ Let $J^* = J \setminus$ (Those intervals in J whose Finish Time $\leq t_\beta$)

⑤ Return $\{I_\beta\} \cup$ COMPUTE-OPT(J^*).

CORRECTNESS

Lemma : $OPT(J) = OPT(J^*) + 1$

Proof : Part 1 : $OPT(J) \leq OPT(J^*) + 1$

Part 2 : $OPT(J) \geq OPT(J^*) + 1$

Q3 :

We have n jobs $J_1 \dots J_n$ s.t. $FT(J_1) \leq \dots \leq FT(J_n)$.

COMPUTE-OPT$(J_1 \dots J_n)$

1. $S = \emptyset$
2. $\alpha = -\infty$
3. For $i = 1$ to n:

   If $(s_i \geq \alpha)$:

   → Add $J_i = [s_i, t_i]$ to set S

   → Set $\alpha = t_i$
4. Return(S)

Main Idea:

In above algo we are iteratively adding to S jobs with earliest FINISH-TIME while ensuring that jobs in S are NON-OVERLAPPING.

Q4:

(a) We will disprove the claim. Consider the graph K_n.

CLAIM 1 : $|VC_{opt}(K_n)| = n-1$

Proof : There are ${}^nC_2$ edges. Any $VC(K_n)$ should have $n-1$ vertices, otherwise we will have uncovered edges.

CLAIM 2 : $|DS_{opt}(K_n)| = 1$

Proof : Take any vertex x in K_n. Then $\{x\}$ is dominating set as entire set $V \setminus \{x\}$ is adjacent to x.

Now, 2-approx algo to compute vertex-cover will return a set "S" of size $\geq n-1$.
It can't be a 2-approximation of $DS_{opt}(K_n)$.

(b)

We will study the case where vertices in $T \geq 3$.
Consider an arbitrary non-leaf node "r" to be root of T.

We will use notation "C" to denote the set of uncovered vertices in T.

Intially C will be entire vertex-set.

COVER(C)

1. If ($C = \emptyset$) then Return $\emptyset$
2. $x \leftarrow$ A node in C of maximum depth.
3. $y \leftarrow$ Parent(x, T)
4. $C' \leftarrow C \setminus \{y$ and neighbors of $y\}$
5. Return $\{y\} \cup$ COVER(C')

Correctness Proof :

Take an instance of Cover(C).

Exchange Lemma: Let x be a node in "C" of maximum depth, and let y = Parent(x, T).

CLAIM : If S is opt solⁿ to COVER(C), then $(S \setminus x) \cup \{y\}$ is also an opt solⁿ.

Intuition for correctness: Similar to correctness of job scheduling covered in Lecture 2.

Implementation : Can be implemented in linear time by scanning tree in BOTTOM-UP manner.

Q5

(a) ALGO ($I_1 \cdots I_n$)

① Sort intervals according to start time in $O(n \log n)$ time, so that $S.T.(I_1) \leq \cdots \leq S.T.(I_n)$.

② $\alpha = \max\limits_{i=1 \text{ to } n} \left(\begin{array}{l} \text{No of intervals} \\ \text{overlapping } s_i \end{array} \right)$

③ Initialize array "LEC" of size n.

④ For $i = 1$ to n

Let C = Intervals in range $I_1 \cdots I_{i-1}$ overlapping with I_i. $|C| \leq \alpha - 1$

Set $LEC(i)$ to be any integer in set $\{1, \dots, \alpha\} \setminus \{ LEC(J) \mid J \in C \}$

⑤ Return LEC

Lemma: $\text{opt-sol}(I_1 \cdots I_n) = \alpha$

Proof: Part 1. $\text{opt-sol}(I_1 \cdots I_n) \leq \alpha$

Part 2. $\text{opt-sol}(I_1 \cdots I_n) \geq \alpha$

Q5(b)

Just increment finish time of all lectures by 30 minutes.