Q1

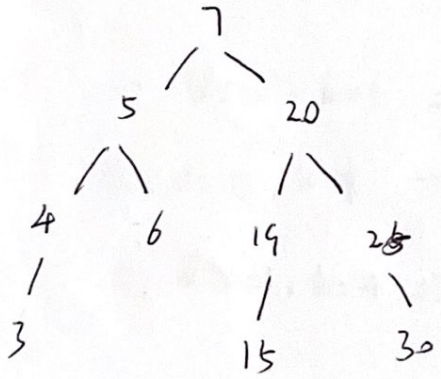
(a) $\Theta(n^5)$

(b) Preorder: 1 2 4 5 8 3 6 7 9 10

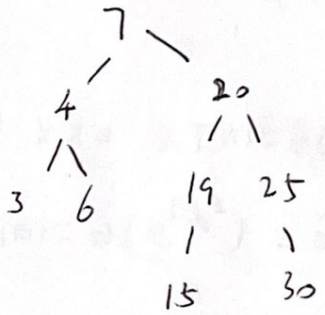
Inorder : 4 2 8 5 1 6 3 9 7 10

Postorder: 4 8 5 2 6 9 10 7 3 1

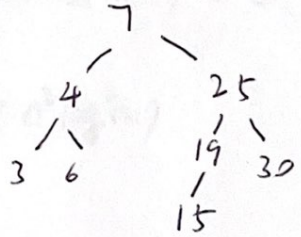
(c) (i)



(ii)



(iii)



(d) (i) ~~True~~ False (ii) True (iii) False

(e) scc : 1 {a, b, c, d, e, f, g, h}

Q2:

(a)

	a	b	c	d	e	f	g	h
pre	1	2	10	3	4	5	13	14
post	12	9	11	8	7	~~6~~	16	15

(b) Tree edge: ab, bd, de, ef, gh, ac

Cross edge: ~~ea~~ cb, cd, ce, cf

Forward : ~~aa~~, ad , af , ~~ae~~

Back ~~ca~~, ~~ea~~, ~~ha~~, ~~gb~~ eg

(c) scc1 : [g] scc3 : [a, e, c, d, b]

scc2 : [h] scc4 : [f]

Q3:

(a) A: $T(n) = 4T(n/2) + \theta(n^2\log n + \log n)$ → $a=4$, $b=2$

B: $T(n) = 2T(n/4) + \theta(\log n)$ → $a=2$ $b=4$

C) $T(n) = 3T(n/9) + \theta(n^{0.5}/\log n)$ → $a=3$ $b=9$

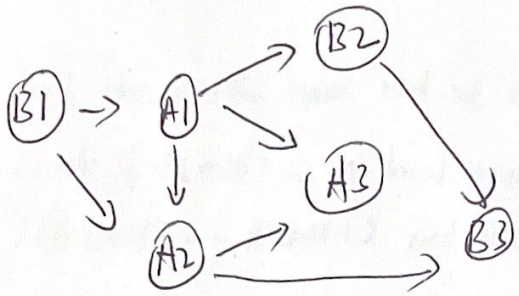
(b): A: $d=2$, $k=1$ ⇒ $\log_2 4 = d$ and $k \geq 0$ ⇒ $T(n) = \theta(n^2\log^2 n)$

B: $d=0$, $k=1$ ⇒ $\log_4 2 > d$ ⇒ $T(n) = \theta(n^{\log_4 2}) = \theta(n^{0.5})$

C: $d=0.51$, $k=-1$ ⇒ $\log_9 3 < d$ ⇒ $T(n) = \theta(f(n)) = \theta(n^{0.51}/\log n)$

(C) $B(\theta(n^{0.5})) \rightarrow C(\theta(n^{0.51}/\log n)) \rightarrow A(n^2\log^2 n)$

Q4: (a)



(b) 1. The graph is a directed graph; Therefore, nodes represent cells and edges delegate dependencies.

2. Perform a Topological sort on the graph.

3. Traverse the sorted list and compute each node's value once when its dependency are resolved.

note: Each edge and node is processed exactly once.

(c) B1 → A1 → B2 → A2 → A3 → B3 · determined by topological sorting.

(d) The running time is a ~~t~~ linear:

1. The topological sort Complexity : O(V+E)

2. Each cell and dependency processed once:
   - No redundant calculations
   - No cycles in the dependency graph (DAG)

Final complexity : O(|V| + |E|)

Q5.

(a) 1. find the middle index mid of the current subarray.

2. check if A[mid] is a local minima:

if A[mid] <= A[mid-1] and A[mid] <= A[mid+1], return A[mid]

3. if the left neighbor is smaller:

- Recur on the left half

4. If the right neighbor is smaller:

- Recur on the right half.

Pseudocode:

```
Function find_minima(A, L, R)
    if L == R;
        return A[L]
    mid = (L+R)/2
    if (mid == L or A[mid] ≤ A[mid-1]) and (mid = end or A[mid] ≤ A[mid+1])
        return A[mid]
    else if mid > L and A[mid-1] < A[mid]:
        return find_minima(A, L, mid-1)
    else:
        return find_minimal A, mid+1, R)
```

(b) $T(n) = T(n/2) + O(1)$ => ~~$a=1, b=2, d=0, k=0 \Rightarrow T(n) = O(\log n)$~~

(c) $a=1, b=2, d=0, k=0 \Rightarrow \log_2 1 = d \Rightarrow$ case 2 : $\Theta(\log n)$

(d)

step	start-end	mid	A[mid]	note
1	1-16	8	5	
2	9-216	12	3	Found