Q1

- d
- a, c d
- a, b, d, e
- c, b

Q2

- c, d
- b, c
- We have to prove that there exits $n > n_0$ such that $T(n) \leq c2^n$. We start with the base cases for $n = 0$ and $n = 1$. $d_0 \leq c \cdot 2^0 = c \cdot 1$, thus c has to be bigger than d_0. $d_1 \leq c \cdot 2^1 = c \cdot 2$, thus c has to be bigger than $\frac{d_1}{2}$. Now for the induction.

  $T(n) = T(n-1) + T(n-2) + d$

  $= c_1 2^{n-1} + c_2 2^{n-2} + d$

  $\leq c_1 2^n + c_2 2^n + 2^n$

  $\leq c2^n$

  For some c larger than $c_1 + c_2 + 1$

Q3

Q3.1

- a (assuming the root is 8), b, ## Q3.2
- c (44, 20, 46, 35, n, 15, 92, 84, 52, 17, 87)
- a (44, 17, 46, 35, n, 15, 92, 84, 52, 20, 87)
- b (44, n, 46, 35, 17, 15, 92, 84, 52, 20, 87)

Q3.3

- 1 (1, 12)

  2 (4, 5)

  3 (3, 6)

  4 (2, 9)

  5 (10, 11)

  6 (7, 8)
- (1 (4 (3 (2 2) 3) (6 6) 4) (5 5) 1)

- Tree edges (1 -> 4, 4 -> 3, 3 -> 2, 4 -> 6, 1 -> 5)

  Back edges (2 -> 4)

  Forward edges (1 -> 6)

  Cross edges (5 -> 2, 5 -> 3)
- (4 -> 3 -> 2 -> 4) everything else is singular components # Q4 ## Q4.1 For question we can either choose to solve it or not to selve it. This leads us to being able to represent each evaluation as a bit string (ie 10110110). Since we have n questions we have n bits which mens 2^n possibles numbers.

Q4.2

```
def PointsQuestions(p, t, T)
    n = p.length
    initialize the matrix M[0..n][0..T] with 0s
    initialize the matrix K[0..n][0..T] with False
    for i = 1 to n
        for j = 1 to T
            if t[i] > j
                M[i,j] = M[i - 1, j]
            else
                if M[i - 1,j] >= p[i] + M[i - 1, j - t[i]]
                    M[i,j] = M[i - 1, j]
                else
                    M[i,j] = p[i] + M[i - 1, j - t[i]]
                    K[i,j] = True
    return M[n,T], K
```

Most things take constant time, although we have a loop from 1 to n, and within that loop we have a loop from 1 to T. Thus the running time will be $O(n \cdot T)$

Q4.3

```
def Questions(p, t, T)
    (P, K) = PointsQuestions(p, t, T)
    j = W
    for i = p.length downto 1
        if K[i,j]
            print i
            j = j - t[i]
```

Q5

```
def All-Pairs-Same-Sequence(W, tau)
    n = W[0].rows
```

```
D(0) = W
for k = 1 to n
    let D(k) be a new 3d array [1..m][1..n][1..n]
    for i = 1 to n
        for j = 1 to n
            for h = 1 to m
                d(k)[h][i][j] = min(d(k - 1), d(k - 1)[h][i][k] + d(k - 1)[h][k][j])
```

This is not going to work

Run time is $\Theta(n^3 \cdot m)$