1. Write a detailed, structured proof that

$$\forall f : \mathbb{N} \rightarrow \mathbb{R}^+, \forall g : \mathbb{N} \rightarrow \mathbb{R}^+, g \in \mathcal{O}(f) \Rightarrow g^2 \in \mathcal{O}(f^2)$$

(where $f^2$ and $g^2$ are defined in the obvious way: $\forall n \in \mathbb{N}, f^2(n) = f(n) \cdot f(n)$, and similarly for $g$).
2. Prove that $T_{\text{BFT}}(n) \in \Theta(n^2)$, where BFT is the algorithm below.

BFT$(E, n)$:
1. $i \leftarrow n - 1$
2. while $i > 0$:
3. $P[i] \leftarrow -1$
4. $Q[i] \leftarrow -1$
5. $i \leftarrow i - 1$
6. $P[0] \leftarrow n$
7. $Q[0] \leftarrow 0$
8. $t \leftarrow 0$
9. $h \leftarrow 0$
10. while $h \leq t$:
11. $i \leftarrow 0$
12. while $i < n$:
13. if $E[Q[h]][i] \neq 0$ and $P[i] < 0$:
14. $P[i] \leftarrow Q[h]$
15. $t \leftarrow t + 1$
16. $Q[t] \leftarrow i$
17. $i \leftarrow i + 1$
18. $h \leftarrow h + 1$

(Although this is not directly relevant to the question, this algorithm carries out a breadth-first traversal of the graph on $n$ vertices whose adjacency matrix is stored in $E$.)
3. Find a tight bound on the worst-case running time of the following algorithm. (This example was started during lecture, but it was not finished.)

    # Precondition: L is a list that contains n > 0 real numbers.
1. max ← 0
2. for i ← 0, 1, . . . , n − 1:
3.     for j ← i, i + 1, . . . , n − 1:
4.         sum ← 0
5.             for k ← i, i + 1, . . . , j:
6.                 sum ← sum + 1
7.             if sum > max:
8.                 max ← sum