1. Find a tight bound on the worst-case running time of the following algorithm.

```plaintext
# 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 + L[k]
7.           if sum > max:
8.               max ← sum
```
2. Prove that $T_{BFT}(n) \in \Theta(n^2)$, where BFT is the algorithm below.

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