1. *Solution 1.* Notice that for any value of j and k, i iterates $k-j+1$ times. So given a specific value of j, since k goes from j to n, the number of times i iterates is

$$1+2+\cdots+(n-j+1)=\frac{(n-j+1)(n-j+2)}{2}\geq\frac{(n-j+1)(n-j)}{2}$$

So

$$\begin{aligned}
\text{total number of iterations} &\geq \sum_{j=1}^{n}\frac{(n-j+1)(n-j+2)}{2}\\
&=\frac{1}{2}\sum_{j=1}^{n}(n^2+n)-(2n+1)j+j^2\\
&=\frac{1}{2}\left(n(n^2+n)-(2n+1)\frac{n(n+1)}{2}+\frac{n(n+1)(2n+1)}{6}\right)\\
&=\frac{n(n^2-1)}{6}
\end{aligned}$$

Now set

$$\frac{n(n^2-1)}{6}\geq cn^3$$

and solve this inequality for n. We get

$$n\geq\frac{1}{\sqrt{1-6c}}.$$

Therefore, if we let $c=1/12$ and $n_0=2$, we have that

$$\text{total number of iterations}\geq cn^3\quad\text{whenever } n\geq n_0.$$

So the runtime is $\Omega(n^3)$.

Solution 2. We have

$$\begin{aligned}
\text{total number of iterations} &= \sum_{j=1}^{n}\sum_{k=j}^{n}\sum_{i=j}^{k}1\\
&\geq\sum_{j=1}^{n/3}\sum_{k=2n/3}^{n}\sum_{i=n/3}^{2n/3}1\\
&\geq\frac{n^3}{27}.
\end{aligned}$$

Therefore, if we let $c=1/27$ and $n_0=1$, we have that

$$\text{total number of iterations}\geq cn^3\quad\text{whenever } n\geq n_0.$$

So the runtime is $\Omega(n^3)$.

Solution 3. Notice that the pseudocode iterates through all possible triplets (j, i, k) such that $j \leq i \leq k$ and such i, j, k each has a value between 1 and n. If we have ignore the duplicate values such as $(2, 2, 3)$, then the total number of such triplets is

$$\binom{n}{3} = \frac{n(n-1)(n-2)}{6}.$$

Set

$$\frac{n(n-1)(n-2)}{6} \geq cn^3.$$

Let $c = 1/12$. This simplifies to

$$n^2 - 6n + 4 \geq 0$$

Now it's easy to see that this inequality holds for all $n \geq 6$. Therefore, if we let $c = 1/12$ and $n_0 = 6$, we have that

$$\text{total number of iterations} \geq cn^3 \quad \text{whenever } n \geq n_0.$$

So the runtime is $\Omega(n^3)$.

2. If $n_0 = 59$, then $10n \log n < n^2$ whenever $n \geq n_0$.

3. All answers below are approximate.

	1 Second	1 Hour	1 Month	1 Century
$\log n$	10^{300000}	10^{10^9}	$10^{9\times10^{11}}$	$10^{9\times10^{14}}$
$\sqrt{n}$	10^{12}	2×10^{19}	9×10^{24}	9×10^{30}
n	10^6	4×10^9	3×10^{12}	3×10^{15}
$n \log n$	6×10^4	10^8	8×10^{10}	7×10^{13}
n^2	10^3	6×10^4	2×10^6	5×10^7
n^3	10^2	2×10^3	2×10^4	2×10^5
2^n	20	32	41	51
$n!$	9	12	15	17

4.
```
reverse(A):
    s = stack()
    for i from 1 to len(A):
        s.push(A[i])
    for i from 1 to len(A):
        A[i] = s.pop()
```

The running time is $2n$ or $O(n)$.