Pseudocode

For each algorithm below, explain why the algorithm is badly expressed and rewrite the algorithm in high-quality pseudocode at a level of detail appropriate for analyzing its performance in terms of time and space.

1. A badly-expressed algorithm to count the number of positives in a list of numbers $A$:

   1. Repeatedly find the next positive number in the list and
   2. increment a counter (which will eventually be returned).

2. A badly-expressed algorithm to multiply the elements of a list:

   1. let $p ← 1$
   2. let $i ← 0$
   3. if $i = |A|$ then jump to 8
   4. let $t ← A[i]$
   5. let $p ← p · t$
   6. let $i ← i + 1$
   7. jump to 3
   8. return $p$

3. A badly-expressed algorithm to determine if two sets, $A$ and $B$, are disjoint:

   1. return $¬∃a ∈ A. a ∈ B$

4. A badly-expressed algorithm to count the edges in a graph:

   1. let $m ← 0$
   2. for $v ∈ G$ do
   3.    foreach $n$ in the neighborhood of $v$ do
   4.    increase $m$ by $\frac{1}{2}$
   5. end
   6. end

5. A badly-expressed algorithm to determine if a graph is complete:

   1. Let $G = (V, E)$. Then $G$ is complete if and only if $|V|^2 = |V| + 2|E|$.

Summations

Compute the following sums, showing your work.

6. $\sum_{i=0}^{2n-1} i$
7. $\sum_{i=0}^{n-1} (5 + i)$
8. $\sum_{i=0}^{n-1} (3^i + 9)$
9. $\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} i$
10. $\sum_{i=0}^{n-1} \sum_{j=i}^{n-1} 2$