Exercises - Algorithmics - Algorithms
SOLUTIONS

Question 1
Give pseudocode for an algorithm to find the largest element in an array. How efficient is your algorithm?

Solution
Data: $A$: an array of numbers

\[ x = -\infty; \]
\[ i = 1; \]

while $A$ has at least $i$ elements do

  if $A[i] > x$ then

    $x = A[i]$;

  end

  $i = i + 1$;

end

return $x$;

This algorithm loops over the array once, which takes $O(|A|)$ time.

Question 2
Give pseudocode for an algorithm to check if a graph has any triangles (cycles of length 3). How efficient is your algorithm?

Solution
Data: $G = (V, E)$: a graph

\[ b = false; \]

for $u$ in $V$ do

  for $v$ in $V$ do

    for $w$ in $V$ do

      if $(u, v) \in E \land (v, w) \in E \land (u, w) \in E$ then

        $b = true$;

      end

    end

  end

end

return $b$;

This algorithm checks all triples of vertices, which is not particularly efficient. Assuming we stored the graph in an adjacency matrix, it runs in $O(|V|^3)$ time.

Question 3
Give pseudocode for an algorithm to check if a graph is connected. How efficient is your algorithm?

Solution
Data: $G = (V, E)$: a graph
pick any $v \in V$;
marked($v$) = true;
$R = \{v\}$;
while $R$ is not empty do
    pick any $v \in R$;
    for $e \in E$ connected to $v$ do
        $w =$ the other vertex of $e$;
        if not marked($w$) then
            marked($w$) = true;
            $R = R \cup \{w\}$;
        end
    end
end
$b =$ true for $v \in V$ do
    if not marked($w$) then
        $b =$ false;
    end
end
return $b$;

This algorithm visits every edge of the graph at most once, so it takes $O(|E|)$ time, provided that we can efficiently access the neighbours of each vertex (that is, if the graph is stored as an adjacency list rather than an adjacency matrix).