Algorithms: Solutions 1

Problem 1

(a) $1 + x + x^2 + x^3 + ... + x^n = \frac{x^{n+1}-1}{x-1}$ (where $x \neq 1$).

We use a proof by induction. Clearly, the equality holds for $n = 1$; we now show that, if it holds for n, then it also holds for $n + 1$:

$$\begin{aligned}
1 + x + ... + x^n + x^{n+1} &= \frac{x^{n+1} - 1}{x - 1} + x^{n+1} \\
&= \frac{x^{n+1} - 1 + x^{n+1} \cdot (x - 1)}{x - 1} \\
&= \frac{x^{n+1} - 1 + x^{n+2} - x^{n+1}}{x - 1} \\
&= \frac{x^{(n+1)+1} - 1}{x - 1}.
\end{aligned}$$

(b) $(1 + 2 + 3 + 4 + ... + n)^2 = 1^3 + 2^3 + 3^3 + 4^3 + ... + n^3$.

We observe that the equality holds for $n = 1$, and apply induction to "step" from n to $n+1$:

$$\begin{aligned}
(1 + 2 + ... + n + (n + 1))^2 &= (1 + 2 + ... + n)^2 + 2 \cdot (1 + 2 + ... + n) \cdot (n + 1) + (n + 1)^2 \\
&= 1^3 + 2^3 + ... + n^3 + 2 \cdot \frac{n \cdot (n + 1)}{2} \cdot (n + 1) + (n + 1)^2 \\
&= 1^3 + 2^3 + ... + n^3 + n \cdot (n + 1)^2 + (n + 1)^2 \\
&= 1^3 + 2^3 + ... + n^3 + (n + 1) \cdot (n + 1)^2 \\
&= 1^3 + 2^3 + ... + n^3 + (n + 1)^3.
\end{aligned}$$

Problem 2

Describe an algorithm that finds the largest and second largest elements of an array $A[1..n]$ using $(n + \lceil \log_2 n \rceil)$ comparisons.

We may think of this problem as a tournament among n chess players. Every number is a player, and every comparison is a game, which reveals the stronger of the two players. We need to plan $(n + \lceil \log_2 n \rceil)$ games that reveal the strongest and second strongest player.

First, we break the players in pairs and make them play, thus revealing the stronger one in each pair. These stronger players enter the second round of the tournament. If the number of players is odd, then one of them enters the second round without playing in the first round. In the second round, we again break the players in pairs and make them play, and then move the winners to the third round. We repeat this operation until finding the strongest player. The total number of games for identifying this player is $(n - 1)$.

The final winner of the tournament has played $\lceil \log_2 n \rceil$ games, and the second strongest player is one of those who have lost to the winner. Thus, there are $\lceil \log_2 n \rceil$ players who have lost to the winner, and we have to find the strongest one among them. We run a smaller tournament among these players, which involves $(\lceil \log_2 n \rceil - 1)$ games and reveals the second strongest player. The total number of games is $(n + \lceil \log_2 n \rceil - 2)$, which is two games fewer than the allowed maximum.