1. The arithmetic progression gives

$$\begin{aligned}\sum_{i=1}^{n} i + n(n+1)/2 &= n((1+n(n+1)/2) + (n+n(n+1)/2))/2 \\ &= n((2+n^2+n)/2 + (3n+n^2)/2)/2 \\ &= (n(2+n^2+n)/2 + n(3n+n^2)/2)/2 \\ &= ((2n+n^3+n^2)/2 + (3n^2+n^3)/2)/2 \\ &= ((2n^3+4n^2+2n)/2)/2 \\ &= (n^3+2n^2+n)/2.\end{aligned}$$

2.

a) The running time is $O(\sqrt{n}) + O(n) + O(1)$ which is $O(n)$.

b) The running time is $O(\sqrt{n}) + O(1) + O(1)$ which is $O(\sqrt{n})$.

3.

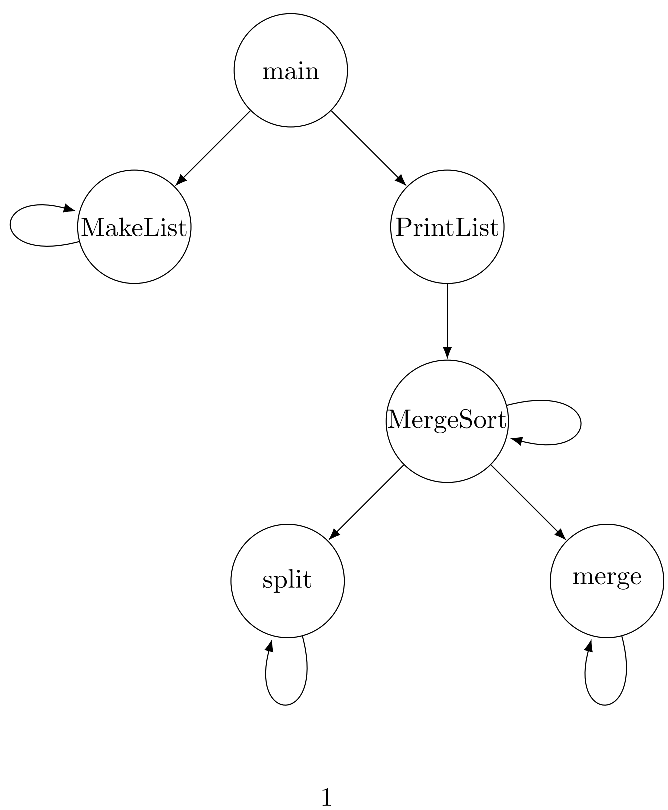
a) The loop goes around $n!$ times taking $O(n)$ time to check the condition and $O(1)$ time in the body. Thus the running time of the function is $O(n \times n!)$.

b) We have $O(n^2)$.

c) We have $O(n^3)$.

d) We have $O(1)$.

4. The merge sort program is recursive. Here is the calling graph.



5. We first analyze `bar`. The running time is $O(n)$ and returns $x + n(n+1)/2$.

We analyze `foo`. The body of the for-loop has as its worst case $i = 1$, thus the running time of the body is $O(n)$. But line (7) has as its condition `i <= bar(n,n)`. The variable `x` in `bar` takes on n and returns $n + n(n+1)/2 = (2n + n^2 + n)/2 = (n^2 + 3n)/2$. Since `n` does not change in `foo`, the loop will go around $(n^2 + 3n)/2$ times, which takes $O(n^2)$ running time. Therefore the loop takes $O(n^3)$ running time. Together with line (9), `foo` takes $O(n^3)$ running time.

We analyze `main`. Line (1) is $O(1)$ and line (2) is $O(n^3)$ and line (3) is $O(n)$. Therefore `main` takes $O(n^3)$ running time.