1...) $W(n) = W(n-1) + 2$ root: 2, level 1: 2 → balanced
Max cost of level: 2
levels: n → $O(n)$

$W(n) = W(n-1) + n^c$, $c \geq 1$ root: n^c, level 1: $(n-1)^c$ → since $c \geq 1$ → balanced
Max cost of level: n^c
levels: n → $O(n \cdot n^c) = O(n^{c+1})$

$W(n) = W(\sqrt{n}) + 1$ root: 1, level 1: 1 → balanced
Max cost of level: 1
levels: $n^{2^{-x}} = 2 \rightarrow x = \log\log n$ $O(\log\log n)$

2) Algorithm A: $W(n) = 5W(n/2) + n$ root: n, level 1: $\frac{5n}{2}$ → leaf dominated
nodes: 5^i
levels: $\log_2 n$ → $O(n^{\log_2 5})$

Algorithm B: $W(n) = 2W(n-1) + c$ root: c, level 1: $2c$ → leaf dominated
nodes: 2^i
levels: n → $O(2^n)$

Algorithm C: $W(n) = 9W(n/3) + n^2$ root: n^2, level 1: $9(\frac{n}{3})^2 = n^2$ → balanced
Max cost of level: n^2
levels: $\log_3 n$ → $O(n^2 \log_3 n)$

Choose Algorithm A because $O(n^{\log_2 5}) < O(n^2 \log_3 n) < 2^n$