Part one
(Mainly use Tree method)

1.
W(n) = 2W(n/3) + 1
Level i contains 2^i nodes
Each node at level i costs 1
So each level cost 2^i
The height of the tree is log3(n), so we have log3(n) levels

So the total cost of the tree is W(n) = $\sum_{i=0}^{log3(n)} 2^i$ = O(n^log3(2))

2.
W(n) = 5*W(n/4) + n
Level i contains 5^i nodes
Each node at level i costs n/4^i
So each level cost 5^i *n / 4^i
The height of the tree is log4(n), so we have log4(n) levels

So the total cost of the tree is W(n) = $\sum_{i=0}^{log4(n)} (5/4)^i * n$ = O(n^log4(5))

3.
W(n) = 7*W(n/7) + n
Level i contains 7^i nodes
Each node at level i costs n/7^i
So each level cost n
The height of the tree is log7(n), so we have log7(n) levels

So the total cost of the tree is W(n) = $\sum_{i=0}^{log7(n)} n$ = O(n*logn)

4.
W(n) = 9*W(n/3) + n^2
Level i contains 9^i nodes
Each node at level i costs n^2/9^i
So each level cost n^2
The height of the tree is log3(n), so we have log3(n) levels

So the total cost of the tree is W(n) = $\sum_{i=0}^{log3(n)} n^2$ = O(n*n*logn)

5.
W(n) = 8*W(n/2) + n^3
Level i contains 8^i nodes
Each node at level i costs n^3/8^i
So each level cost n^3
The height of the tree is log2(n), so we have log2(n) levels

So the total cost of the tree is W(n) = $\sum_{i=0}^{log2(n)} n$^3 = O(n*n*n*logn)

6.
W(n) = 49*W(n/25)+n^(3/2)*logn
a = 49, b = 25
b^1.5 = 125 > a
Therefore W(n) = O(n^(3/2)*logn)

6.
W(n) = 49*W(n/25)+n^(3/2)*logn
C(root) = n^(3/2)*logn
C(level 1) = 49*(n/25)^(3/2)*log(n/25) = 49/125 * n^(3/2) * (logn – log25)
So if a <- 125 / 49
n^(3/2)*logn >= 125 / 49 * 9/125 * n^(3/2) * (logn – log25)
n^(3/2)*logn >= n^(3/2) * (logn – log25)
Therefore cost: O(n^(3/2) * logn)

7.
W(n) = W(n-1)+2 =W(n-2)+4 =O(n)

8.
It is balanced and there are n levels
W(n) = O(n*n^c) = O(n^(c+1))

9.
W(n) = W(sqrt(n)) +1
Level i contains 1 nodes
Each node at level i costs 1
So each level cost 1
The height of the tree is log(n)*log(n), so we have log(n)*log(n) levels
So the total cost of the tree is W(n) = $\sum_{i=0}^{log(n)*\log(n)} 1$ = O(log(n)*log(n))

Part two:

For Algorithm A, W(n) = 5 * W(n/2) +n
Level i contains 5^i nodes
Each node at level i costs n/2^i
So each level cost (5/2)^i * n
The height of the tree is log5(n), so we have log5(n) levels
So the total cost of the tree is W(n) = $\sum_{i=0}^{log5(n)} (\frac{5}{2})^i * n$ = O(n^(log2(5)))

For Algorithm B, W(n) = 2*W(n-1) + 1 = O(2^n)

For Algorithm C, W(n) = 9 * W(n/3) + n^2
Level i contains 9^i nodes
Each node at level i costs n^2/9^i
So each level cost n^2
The height of the tree is log3(n), so we have log3(n) levels

So the total cost of the tree is W(n) = $\sum_{i=0}^{log3(n)} n \,\hat{}2$ = O(n*n*logn)

Therefore we should choose algorithm C