Analysis of the Bubble Sort Algorithm

for (int i = 0; i < n-1; i++)
    for (int j = 0; j < n-1; j++)
        if (x[j] > x[j+1])
            // Swap x[j] with x[j+1];
Analysis of the Bubble Sort Algorithm

for (int i= 0; i < n-1; i ++)
    for (int j = 0; j < n-1; j ++)
        if (x[j] > x[j+1])
            // Swap x[j] with x[j+1];

Suppose:

- \( t_1 \) = time to set up a for loop
- \( t_2 \) = time to increment the loop variable and test it against the limit
- \( t_3 \) = time to compare two elements
- \( t_4 \) = time to switch them.
- \( p \) = probability two items need to be switched
Analysis of the Bubble Sort Algorithm

for (int i = 0; i < n-1; i++)
    for (int j = 0; j < n-1; j++)
        if (x[j] > x[j+1])
            // Swap x[j] with x[j+1];

Suppose:

- $t_1 =$ time to set up a for loop
- $t_2 =$ time to increment the loop variable and test it against the limit
- $t_3 =$ time to compare two elements
- $t_4 =$ time to switch them.
- $p =$ probability two items need to be switched

Overall time $= t_1 + (n-1) (t_2 + t_1 + (n-1) (t_2 + t_3 + pt_4))$
$= (t_2 + t_3 + pt_4) \times n^2 + (t_1 - t_2 - 2t_3 - 2pt_4) \times n + (t_3 + pt_4)$
Analysis of the Bubble Sort Algorithm

for (int i = 0; i < n-1; i++)
    for (int j = 0; j < n-1; j++)
        if (x[j] > x[j+1])
            // Swap x[j] with x[j+1];

Suppose:

- t1 = time to set up a for loop
- t2 = time to increment the loop variable and test it against the limit
- t3 = time to compare two elements
- t4 = time to switch them.
- p = probability two items need to be switched

Overall time = t1 + (n-1) (t2 + t1 + (n-1) (t2 + t3 + pt4))
              = (t2 + t3 + pt4) * n^2 + (t1 - t2 - 2t3 - 2pt4) * n + (t3 + pt4)

This has the form: c1 * n^2 + c2 * n + c3
where c1 = t2 + t3 + pt4;  c2 = t1 - t2 - 2t3 - 2pt4;  c3 = t3 + pt4