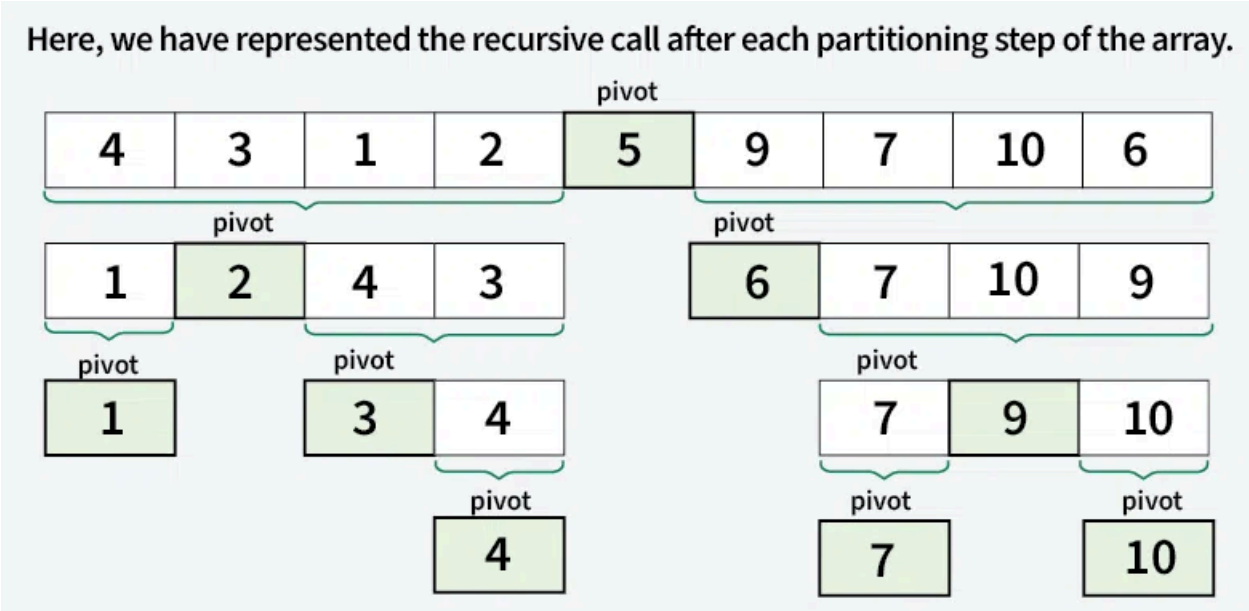
Answer the following questions:

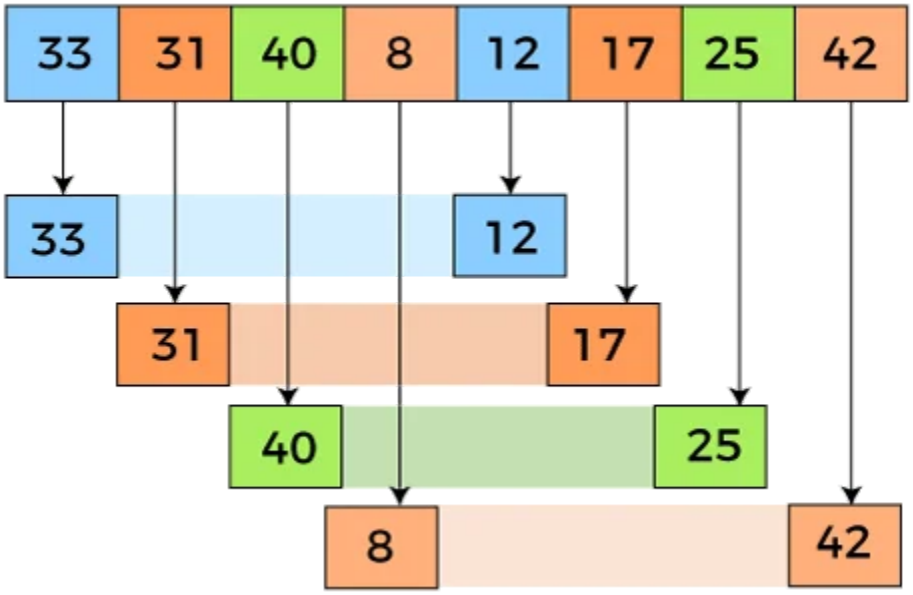
1. Explain the quick sort, shell sort and merge sort types of sorting algorithms.

When using quick sort we have to include how to choose the best pivot, because choosing a pivot can determine how fast the algorithm is. By choosing a good pivot we can deduce the big O notation to be O(n log n). This is the ideal notation for the quick sort, but when the pivot choice is bad it can turn to O(n^2). On how the quick sort according to (GeeksforGeeks, 2025), QuickSort is a sorting algorithm based on the Divide and Conquer that picks an element as a pivot and partitions the given array around the picked pivot by placing the pivot in its correct position in the sorted array. It works on the principle of divide and conquer, breaking down the problem into smaller sub-problems.

GeeksforGeeks. (2025, August 8). Quick sort. GeeksforGeeks. https://www.geeksforgeeks.org/dsa/quick-sort-algorithm/

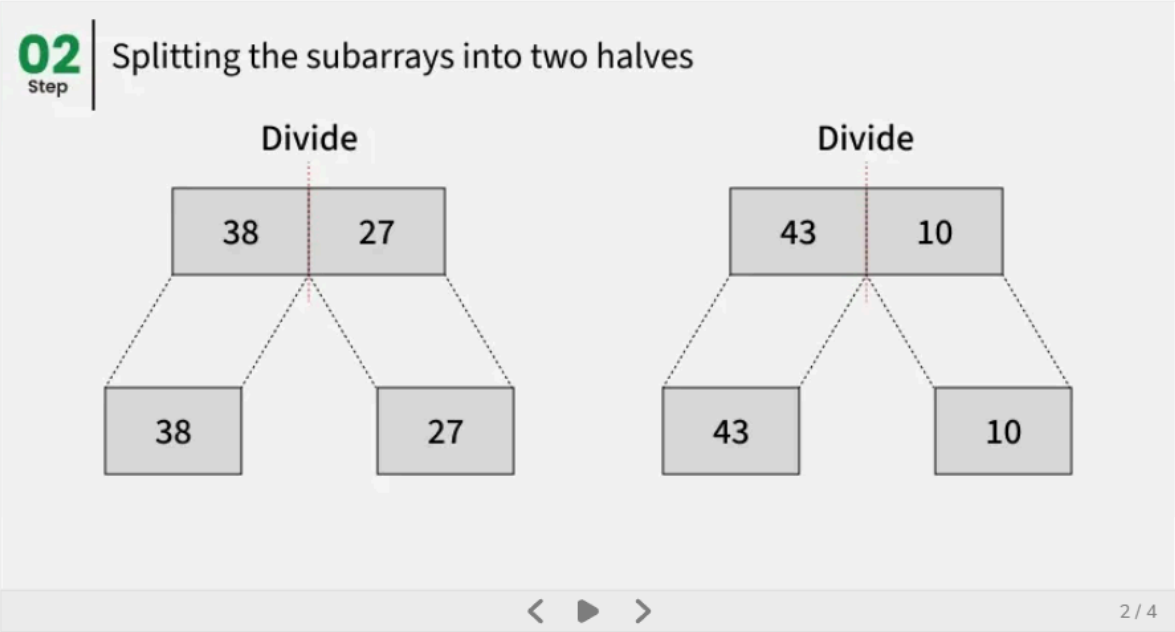


The shell sort is just the enhanced version of the insertion sort. On how it works we just need to divide the array into 2 sub arrays, making it a 2 array now. When the array is odd we can round it down for example if we have 7 elements divided to 2 making it 3.5 when we round down it will be 3. By doing this we can reduce the gap which is the point to the element that we want to swap. Then after doing all this we can now use the insertion sort, after this if the array is still not sorted we will just have to repeat all the steps until the array is sorted.

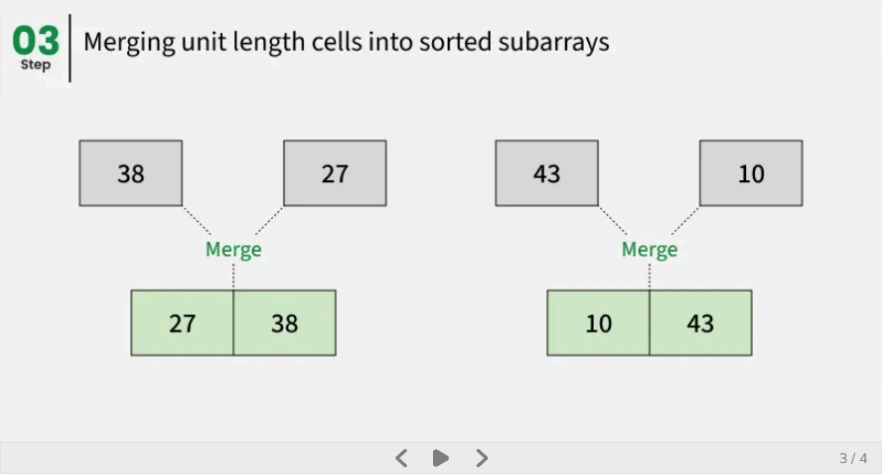


Shell Sort - Expert mentoring, Customized solutions. (n.d.). https://venkys.io/articles/details/shell-sort

Merge sort is just like quick sort with Divide and conquer. The difference is that it has the merge just like in the algorithm name. This actually merges all the elements into the array. By the way I described it, we have to first divide the array into 2. For example we have an array of 6 then divide it by 2 we have 3 on each side. After that we have to divide it again until we are left by 1 single element



After this we now have to compare the single array on each side, after this step we now have to merge it



Since the each side is sorted, we just now have to compare it again then merge the sorted array.

GeeksforGeeks. (2025b, September 23). Merge sort. GeeksforGeeks. https://www.geeksforgeeks.org/dsa/merge-sort/

2. Give simple sample programs in C++ that uses the above sorting algorithms. Use a user input of 10 integer values in your example elements in an array to be sorted. Explain how the programs work.

```cpp
#include <iostream>

using namespace std;


void quickSort(int arr[], int low, int high) {

    if (low < high) {

        int pivot = arr[high]; // Choose last element as pivot

        int i = low - 1;


        for (int j = low; j < high; j++) {

            if (arr[j] < pivot) {

                i++;

                swap(arr[i], arr[j]); // Move smaller element to left

            }

        }

        swap(arr[i + 1], arr[high]); // Place pivot in correct position

        int pi = i + 1;


        quickSort(arr, low, pi - 1); // Sort left part

        quickSort(arr, pi + 1, high); // Sort right part

    }
```

```
}


int main() {
    int arr[10];
    cout << "Enter 10 integers:\n";
    for (int i = 0; i < 10; i++) cin >> arr[i];

    quickSort(arr, 0, 9);

    cout << "Sorted array (Quick Sort): ";
    for (int i = 0; i < 10; i++) cout << arr[i] << " ";
    return 0;
}
```

Explanation:
Since the array is supposed to be 10 elements, by the definition of the quick sort of divide and conquer. We have to first divide the array by 2 to make a 2 sub array. Then we will select the pivot as we can now do some if statements that include elements that is less than and greater than the pivot.