Sort An Array without Recursion

```
class Solution {
public:
    vector<int> merge(vector<int> A, vector<int> B){
        int ASize = A.size(), BSize = B.size();
        vector<int> merged(ASize + BSize);
        int i = 0, j = 0, k = 0;
        while (i < ASize && j < BSize){
            if (A[i] <= B[j]){
                merged[k] = A[i];
                i++;
            }
            else{
                merged[k] = B[j];
                j++;
            }
            k++;
        }
        while (i < ASize){
            merged[k] = A[i];
            i++;
            k++;
        }
        while (j < BSize){
            merged[k] = B[j];
            j++;
            k++;
        }
        return merged;
    }
    vector<int> sortArray(vector<int>& nums) {
        queue<vector<int>> q;
        for (int i = 0; i < nums.size(); i++){
            q.push(vector<int>{nums[i]});
        }

        while (q.size() != 1){
            vector<int> q1 = q.front(); q.pop();
            vector<int> q2 = q.front(); q.pop();
            q.push(merge(q1, q2));
        }

        return q.front();
    }
};
```