254 Project Bitonic Sort

Students:

مهند الرشيد 439101298

مشهور البكر 439102235

Test runs:

Run 1:

```
D:\Work\University files\4th Year\453\Project>solution
Enter the number of elements of the array (inputs not in range 2^x will be zero padded): 5
Number of elements is 2^3 (=8) padded with 3 zeros totalling a size of 32
Do you want to fill the array by hand? (y\n): y
Please input a number for the 0th placement: 1
Please input a number for the 1th placement: 9
Please input a number for the 2th placement: 2
Please input a number for the 3th placement: 8
Please input a number for the 4th placement: 3

Array construction complete. Press enter to begin sort. 7


Index           Value
0               0
1               0
2               0
3               1
4               2
5               3
6               8
7               9
```

Run 2:

```
D:\Work\University files\4th Year\453\Project>solution
Enter the number of elements of the array (inputs not in range 2^x will be zero padded): 16
Number of elements is 2^4 (=16) padded with 0 zeros totalling a size of 64
Do you want to fill the array by hand? (y\n): n

Array construction complete. Press enter to begin sort.


Index           Value
0               2166
1               4006
2               8282
3               9339
4               10114
5               10487
6               10496
7               16037
8               16189
9               16495
10              20027
11              20723
12              21178
13              21950
14              23466
15              24967
```

Code:

```c
#include <time.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

void fillRandom(int *a, int nb){
    for(int i=0; i < nb; i++){
        a[i] = rand() % 100000;
    }
}

void fillUser(int *a, int nb, int last){
    for(int i=0; i < nb; i++){
        if(i < last){
            printf("Please input a number for the %dth placement: ", i);
            scanf("%d", a + i);
        }
```

```
        else
            a[i] = 0;
    }
}

 __device__ void swap(int *a, int *b){
    int temp = *a;
    *a = *b;
    *b = temp;
 }

__global__ void bitonicSort(int *a, unsigned long nb, int step, int stage) {
    unsigned int seqL = pow(2, step);
    unsigned int N = seqL / pow(2, stage - 1);
    unsigned int shift = N / 2;
    short working = threadIdx.x % N < shift;
    short ascending = threadIdx.x / seqL % 2 == 0;

    if(working)
        // printf("ThreadIdx: %d\tAscending: %d\n", threadIdx.x, ascending);
        if(ascending){
            if(a[threadIdx.x] > a[threadIdx.x + shift] == 1)
                swap(a + threadIdx.x, a + threadIdx.x + shift);
        }
        else
            if(a[threadIdx.x] < a[threadIdx.x + shift] == 1)
                swap(a + threadIdx.x, a + threadIdx.x + shift);
}


int main(void) {
    srand(time(NULL));

    int *a, *d_a;
    unsigned long nb, size;

    printf("Enter the number of elements of the array (inputs not in range 2^x will be zero
padded): ");
    scanf("%lu", &nb);

    int exp = ceil(log(nb)/log(2));
    unsigned long newNb = pow(2, exp);
    unsigned long zeros = newNb - nb;
```

```
    nb = newNb;
    size = nb * sizeof(int);
    printf("Number of elements is 2^%d (=%lu) padded with %lu zeros totalling a size of
%lu\n", exp, nb, zeros, size);

    a = (int *)malloc(size);

    printf("Do you want to fill the array by hand? (y\\n): ");
    getchar(); // Flush
    int answer = getchar();
    answer = answer == 121 || answer == 89; // Check if answer is y or Y

    if(answer)
        fillUser(a, nb, nb - zeros);
    else
        fillRandom(a, nb);

    printf("\nArray construction complete. Press enter to begin sort. ");
    getchar(); getchar();

    cudaMalloc((void **)&d_a, size);
    cudaMemcpy(d_a, a, size, cudaMemcpyHostToDevice);

    for(int step=1; step <= exp; step++)
        for(int stage=1; stage <= step; stage++)
            bitonicSort<<<1, nb>>>(d_a, nb, step, stage);

    cudaMemcpy(a, d_a, size, cudaMemcpyDeviceToHost);

    // Print results
    printf("\n\nIndex\t\tValue\n");
    for(unsigned long i=0; i < nb; i++)
        printf("%lu\t\t%d\n", i, a[i]);

    // Cleanup
    free(a);
    cudaFree(d_a);

    return 0;
}
```

Thank you very much.