254 Project Bitonic Sort

Students:

مهند الرشيد 439101298

مشهور البكر 439102235

Test runs:

Run 1:

Text

Description automatically generated

Run 2:

Text

Description automatically generated

Code:

#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.