

Hamid Suha, Gerard Lagos, Dylan Gunter

Project 2 Report

CPSC 479-02

Professor Doina Bein

Emails:

Hamid Suha - hsuha@csu.fullerton.edu

Gerard Lagos - gerardlagos01@csu.fullerton.edu

Dylan Gunter - dgunter@csu.fullerton.edu

Summary: For this project we chose the topic of parallel sorting by creating a heap sort algorithm while programming in CUDA.

How to run code:

1. Open command prompt
2. Connect to CSUF servers using command `ssh -l (username) aries.ecs.fullerton.edu`
3. Create a directory where you can create files using command `mkdir (name)`. (emacs (filename) on mac)
4. Copy code and paste into file by right clicking or using `ctrl+v` (command `c`, command `v` on mac)
5. Save and exit file, compile code with `nvcc (filename).cu`
6. Run program with command `./a.out`

Pseudocode:

```
function heapify(arr, n, i)
    largest = i
    left = 2 * i + 1
    right = 2 * i + 2

    if left < n and arr[left] > arr[largest]
        largest = left

    if right < n and arr[right] > arr[largest]
        largest = right

    if largest != i
        swap(arr[i], arr[largest])
        heapify(arr, n, largest)

function heapSort(arr)
    n = length(arr)

    // Build max heap
    for i = n / 2 - 1 to 0 step -1
        heapify(arr, n, i)

    // Extract elements one by one from the heap
    for i = n - 1 to 1 step -1
        swap(arr[0], arr[i])
        heapify(arr, i, 0)

procedure main()
    // Initialize an array with unsorted values
    inputArray = {12, 11, 13, 5, 6, 7, 2, 9, 8, 10}

    // Perform Heap Sort
    heapSort(inputArray)

    // Print the sorted array

    print("Sorted Array: ", inputArray)
```

Screenshots:

Output 1:

```
dgunter@prudence:~/479$ ./a.out
Unsorted array: 25 33 70 102 56 9 11 32
Sorted Array: 9 11 32 33 25 56 70 102
dgunter@prudence:~/479$
```

Output 2:

```
^[[Adgunter@prudence:~/479$ emacs proj2.cu
dgunter@prudence:~/479$ nvcc proj2.cu
dgunter@prudence:~/479$ ./a.out
Unsorted array: 13 5 6 7 2 9 8 10
Sorted Array: 2 5 6 7 8 9 10 13
dgunter@prudence:~/479$
```