

GPGPU Assignment #1

TA: Yu Sheng Lin

Instructor: Wei Chao Chen

February 29, 2016

1 Goals

You have to

1. Learn how to implement some parallel algorithms on GPU.
2. Learn how to use Thrust library.
3. Be creative!

in this assignment.

2 Requirements

This assignment is split into 3 parts. Part I and II is closed, and part III is opened.

2.1 Part I: Count the Position in Words (60pts)

In this part you are required to count the position of characters in a word. Listing 1. provides a sample input and output.

```
1 aaa aa  a aaaaa  aaaaaa (input)
2 123012001012345000123456 (output)
```

Listing 1: Example: Count the Position in Words.

You may easily come up with an $O(n)$ sequential algorithm and an $O(nk)$ parallel algorithm, where n is the length of the input and k is the maximum length of a word. However in this assignment k is very large deliberately, so can you come up with an $O(n \ln k)$ algorithm? We provide some hints about the algorithm in a separate PDF, and you can decide whether to read them liberally.

The input is generated pseudo-randomly, and you can assume that $k = 500$, $n \approx 2 \times 10^7$. Besides, the input only contains characters [a-z] and we use linebreak '\n' as the spaces.

You have to implement a function whose signature is Listing 2.. All pointers are device pointers and `text_size` is the n .

We also provide some hints not relating to the algorithm itself.

- A kernel may start before previous kernels have finished. To prevent this, insert `cudaDeviceSynchronize()` between them.
- `blockDim.x` cannot exceed 131072 if you don't use `-arch sm_30` compile flag.

```
1 void CountPosition(const char *text, int *pos, int text_size);
```

Listing 2: Example: Count the Position in Words.

2.2 Part II: Find the Heads (40pts)

Using the result of part I, you can find all heads of the characters, namely the position of 1's. You can only use `thrust::*` functions and CUDA API in this part. To be more precise, `__global__` functions are not allowed. We will give you correct answer of part I, so you can do this part even if you cannot finish part I.

Here are some hints:

- Of course you need the document <https://thrust.github.io/doc/modules.html>
- I have already included some necessary headers.
- I used " " (white text).

2.3 Part III: Be Creative! (20pts bonus)

With the results you got in previous parts, please do something interesting.

If you have no ideas about what to do, then you can try the "quite hard" part in assignment #0. I think that it's not that hard now (hopefully).

```
1 ./a.out test.txt
2 Olerm pium odolr ist mate, ocsnceetutr dapisiicgn leti. Uq
3 ...
```

Listing 3: Swap all pairs in all words.

The points you will get is based on your creativity, and you will get 10pts if you implement the "Swap all pairs" example.

3 Submission

- The submission deadline is [TODO].
- For part I and II, [TODO].
- The efficiency of part I will also be considered when grading, and pure CPU implementation is counted as cheating.
- For part III, [TODO].
- The compile flags are `--std=c++11 -arch sm_30 -O2`.
- Please also refer to assignment #0 for more details :P.