

3. Given the following CUDA program, how can you improve the performance? Justify your answer. [10 pts]

```
int main() {
    float h_a[1024], h_b[1024];
    ...
    /*
     * assume appropriate cudaMalloc and cudaMemcpy called to
     * create and initialize d_a and d_b.
     */
    compute<<<16, 256>>>(d_a,d_b);
}

__global__ compute(float *a, float *b) {
    const int tid = threadIdx.x + blockIdx.x * blockDim.x;

    if(tid % 2 == 0){
        operation_even(a, b);
    }
    else {
        operation_odd(a, b);
    }
}
```

Your answer:

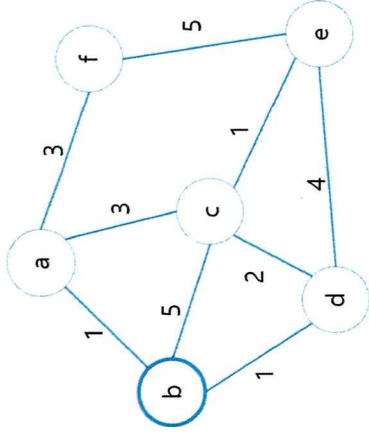
Warp divergence

even 과 odd를 2개의 kernel로 분리한다

4. Prim's algorithm is a greedy algorithm that finds MST. Given the following graph, we run three processes to find its MST in parallel starting from b, then d, a, e, and f.

(a) How will you partition given input numbers? [5 pts]

Your answer:



	a	b	c	d	e	f
d[]	1	0	5	1	∞	∞
a	0	1	3	∞	∞	3
b	1	0	5	1	∞	∞
c	3	5	0	2	1	∞
d	∞	1	2	0	4	∞
e	∞	∞	1	4	0	5
f	2	∞	∞	∞	5	0

(b) Describe four necessary steps in parallel Prim's algorithm. [10 pts]

Partition inputs first.

For each iteration,

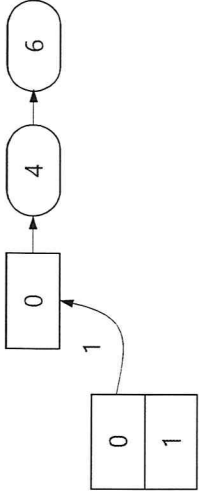
Step 1: ~~Global min~~ P: 가 local min d:[u]를 계산한다

Step 2: Global min 계산은 global P_o에 All - to - One reduction을 한다.

Step 3: 'u'를 one - to - All Broadcast 한다

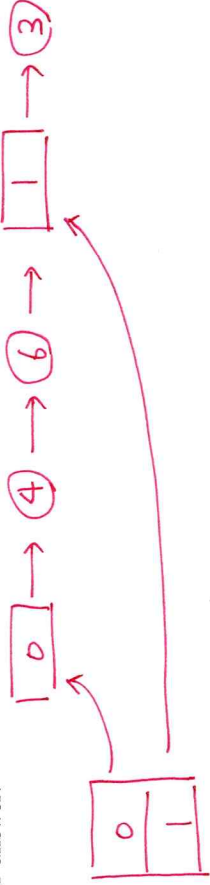
Step 4: d[u]의 local update를 한다

5. The following figure shows a lock-free recursive split-ordering hash table. The capacity of each bucket is 2. Keys are guaranteed to be smaller than 8, hence only 3 bits are needed for keys. We insert a sentinel node when the first item is inserted into a bucket.



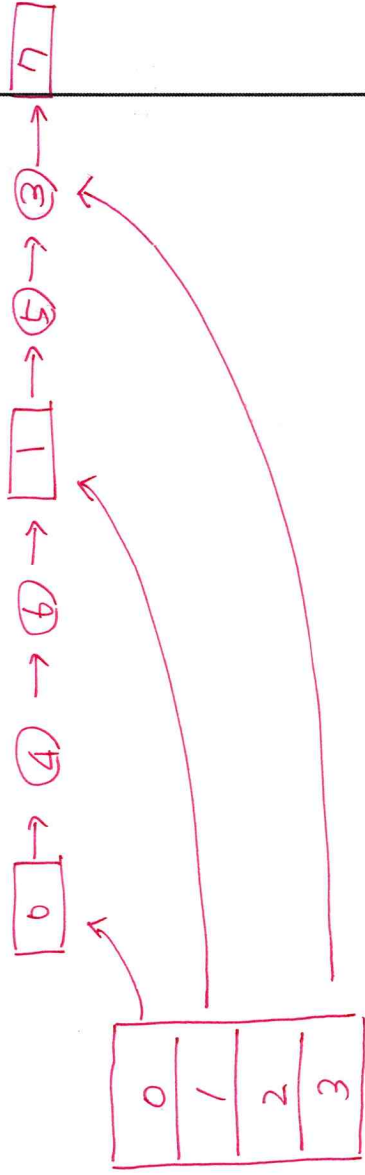
- (a) Illustrate how the split-ordering hash table will be changed after we insert a new key 3. [5 pts]

Your answer:



- (b) Illustrate how the split-ordering hash table will be changed from (a) after we insert two more keys: 5 and 7. [10 pts]

Your answer:



Student ID	Name

Q1	Q2	Q3	Q4	Q5	Total

For Instructor/TA only,

Academic Honor Pledge

I promise or affirm that I will not at any time be involved with cheating or plagiarism while enrolled as a student in Operating Systems class at Sungkyunkwan University.

I understand that violation of this code will result in penalties as severe as Indefinite suspension from the University

Your Signature: _____

1. True or False: Choose the correct answer [40 pts]

- | | T | F |
|--|-------------------------------------|-------------------------------------|
| (a) SIMD is a parallel programming model for distributed memory. | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| (b) MPI is an SPMD programming library. | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| (c) MPI is an SPMD programming library. | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| (d) Data stored in GPU local memory is visible only to a single CUDA thread. | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| (e) Data stored in GPU shared memory is visible to every CUDA thread in the GPU device. | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| (f) Access to the shared memory in the same bank by two CUDA threads in different warps creates a bank conflict. | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| (g) In block-stripe analysis of PageRank, each stripe contains only source nodes in the corresponding block of the new rank vector. | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| (h) One of the major performance bottlenecks in Hadoop compared to Spark is disk I/O for shuffling. | <input checked="" type="checkbox"/> | <input type="checkbox"/> |

2. Do the following codes have a loop-carried dependence? If so, identify the type of each dependence (i.e., true dependence, anti dependence, or output dependence). If possible, remove the dependence.

(a) [10 pts]

```
for (i = 0; i < N - 2; i++) {  
  ① → a[i] += a[i + 2] + 5;  
  ② → x += a[i];  
}
```

Your answer:

anti - dependence

① 과 ② 를 분리한다

(b) [10 pts]

```
for (i = 0; i < N - 1; i++) {  
  ① → a[i] = a[i] + b[i];  
  ② → b[i+1] = c[i] + d[i];  
}
```

Your answer:

True dependence

① 를 loop 밖으로 빼낸다

①
for {
 ② ①
}