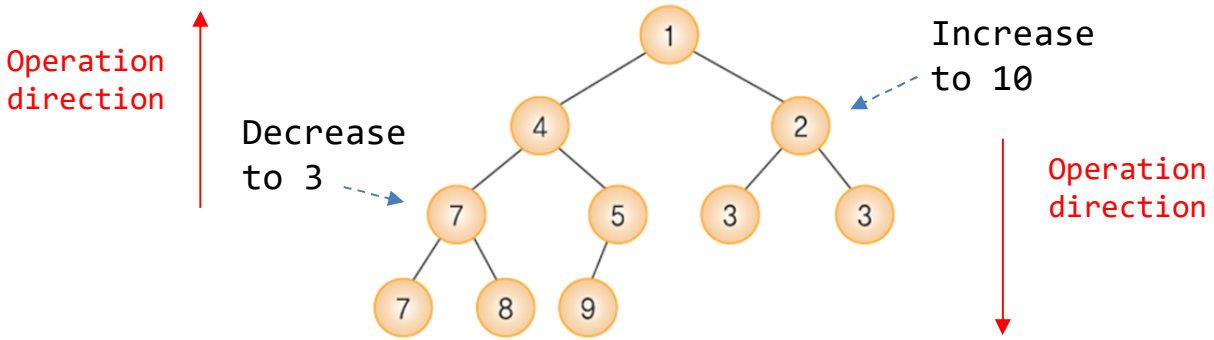
1. (Programming: **30 points**)
Implement the ‘decrease-key’ and ‘increase_key’ in the min-heap, as explained in p53 of ‘DS-Lec10-Graph’.
(Hint: refer to the insert operation in the min heap)
Test your code using the following input.




```
//decrease the element i’s value to ‘key’
Decrease_key_min_heap(A, i, key)
{
   if key >= A[i]
      error “new key is not smaller than current key”;
   //Implement your code below.
   ...
}
```

```
//Increase the element i’s value to ‘key’
Increase_key_min_heap(A, i, key)
{
   if key <= A[i]
      error “new key is not larger than current key”;
   //Implement your code below.
   ...
}
```

2. (Programming: **70 points**)
In p55-56 of 'DS-Lec10-Graph', Prim algorithm was implemented using an unsorted array `'dist'`. Revise this code by referring to p51.

- Use the min heap, not the unsorted array.

- Specify the parent and child relation inside the code.

- Use the following two functions.

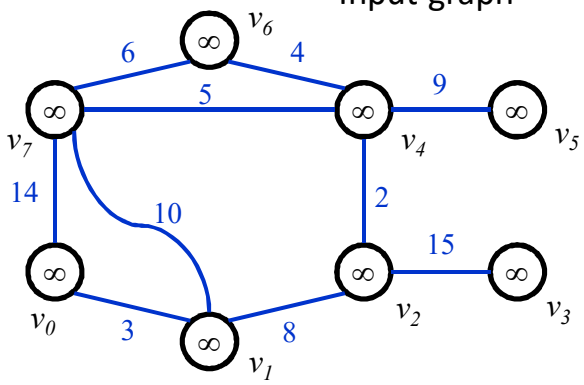
`'build_min_heap'`: use this at `'Insert all vertices into the priority queue Q'`
`'delete_min_heap'`: use this at `'Extract_Min(Q)'`
`'Decrease_key_min_heap'`: use this at `'dist[v] ← weight[u][v]'`

2. (Programming: **70 points**) – Cont.

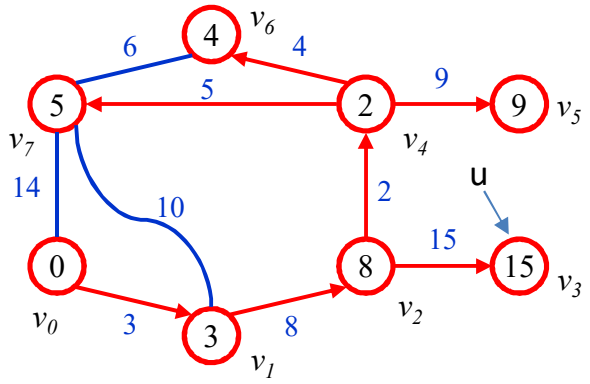
- Use the following input graph.
- Implement ‘`print_prim`’ that prints out as below.
  Hint: revise ‘preorder’ in p22 of ‘DS-Lec07-Tree’.
  Note that Prim MST is not a binary tree.
  Consider how to use parent-child relation.

```
#define MAX_VERTICES 8
#define INF 1000L
int weight[MAX_VERTICES][MAX_VERTICES] =
{ { 0,3,INF,INF,INF,INF,INF,14 },
{ 3,0,8,INF,INF,INF,INF,10 },
{ INF,8,0,15,2,INF,INF,INF },
{ INF,INF,15,0,INF,INF,INF,INF },
{ INF,INF,2,INF,0,9,4,5 },
{ INF,INF,INF,INF,9,0,INF,INF },
{ INF,INF,INF,INF,4,INF,0,6 },
{ 14,10,INF,INF,5,INF,6,0 } };
```

Input graph



Output MST



‘print_prim’ result

```
Vertex 0 -> 1          edge: 3
Vertex 1 -> 2          edge: 8
Vertex 2 -> 3          edge: 15
Vertex 2 -> 4          edge: 2
Vertex 4 -> 5          edge: 9
Vertex 4 -> 6          edge: 4
Vertex 4 -> 7          edge: 5
```