**Bài 100:**

**Pseudo code**

Node (int value, int list\_num, int index);

Comp (gán toán tử min-heap cho priority-queue);

Program findMinimumRange

Input: vector<vector<int>> list

Output: pair<int, int>

{

If lists.size()==0 then return {-1,-1};

M:=lists.size();

high:=INT\_MIN;

pair<int, int> p ={0,INT\_MAX};

priority\_queue<Node, vector<Node>, comp> pq;

for i := 0 to M do {

if lists[i].size() == 0 then return {-1, -1};

pq.push({lists[i][0], i, 0});

high = max(high, lists[i][0]);

}

while (true) {

int low = pq.top().value;

int i = pq.top().list\_num;

int j = pq.top().index;

pq.pop();

if (high – low) < (p.second - p.first) then p := {low, high};

if j == lists[i].size() – 1 then return p;

pq.push({lists[i][j + 1], i, j + 1});

high = max(high, lists[i][j + 1]);

}

};

int main()

{

lists = { { 3, 6, 8, 10, 15 },

{ 1, 5, 12 },

{ 4, 8, 15, 16},

{ 2, 6 } };

pair<int, int> p = findMinimumRange(lists);

cout <<p;

return 0;

}