Best First Search

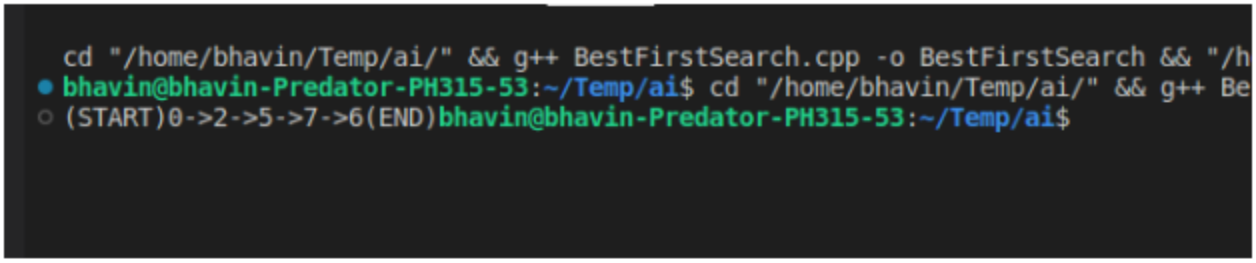
Name: Bhavin Patil
Roll No.: 66

Code:

```cpp
#include <bits/stdc++.h>
using namespace std;

typedef pair<int, int> p;
vector<vector<p>> graph;

void BestaddValuestoGraph(int root, int child, int Hvalue)
{
   graph[root].push_back(make_pair(Hvalue, child));
   graph[child].push_back(make_pair(Hvalue, root));
}


void BestFS(int start, int end, int nodes)
{
   vector<bool> visited(nodes, false);
   priority_queue<p, vector<p>, greater<p>> openpq;
   openpq.push(make_pair(0, start));
   int currStart = start;
   visited[currStart] = true;
   while (!openpq.empty())
   {
       int x = openpq.top().second;
       if (x == start)
           cout << "(START)" << x << "->";
       else if (x == end)
       {
           cout << x << "(END)";
           break;
       }
       else
           cout << x << "->";
```

```
        openpq.pop();
        for (int i = 0; i < graph[x].size(); i++)
        {
            if (!visited[graph[x][i].second])
            {
                visited[graph[x][i].second] = true;
                openpq.push(make_pair(graph[x][i].first,
graph[x][i].second));
            }
        }
    }
}

int main()
{
    int numOfNodes = 8;
    graph.resize(numOfNodes);

    //Best First Search Graph
    BestaddValuestoGraph(0, 1, 32);
    BestaddValuestoGraph(0, 2, 25);
    BestaddValuestoGraph(0, 3, 35);
    BestaddValuestoGraph(1, 4, 19);
    BestaddValuestoGraph(2, 4, 19);
    BestaddValuestoGraph(2, 5, 17);
    BestaddValuestoGraph(3, 5, 17);
    BestaddValuestoGraph(4, 6, 10);
    BestaddValuestoGraph(5, 7, 0);
    BestaddValuestoGraph(6, 7, 0);

    int start = 0;
    int end = 6;
    BestFS(start, end, numOfNodes);

    return 0;
}
```

Output:


cd "/home/bhavin/Temp/ai/" && g++ BestFirstSearch.cpp -o BestFirstSearch && "/h
bhavin@bhavin-Predator-PH315-53:~/Temp/ai$ cd "/home/bhavin/Temp/ai/" && g++ Be
(START)0->2->5->7->6(END)bhavin@bhavin-Predator-PH315-53:~/Temp/ai$