Abdullah Yalçın 308520189001

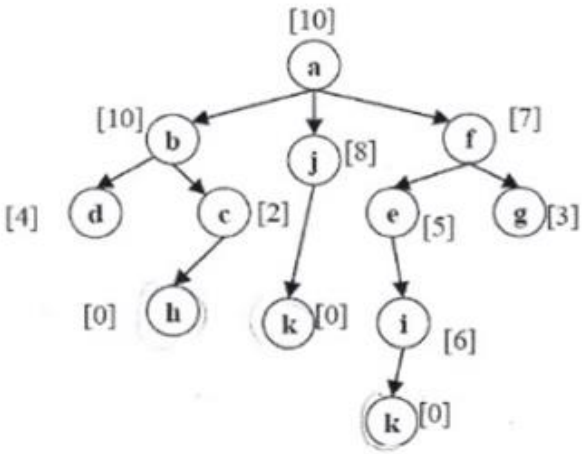
Simulated Annealing

In this example, I have used graph and my node classes and their attributes. I have implemented annealing search algroithm.

```
public List<Node<T>> SimulatedAnnealing(PriorityQueue<Node<T>, int> openList, int temperature)
{
    Stack<Node<T>> stack = new Stack<Node<T>>();

    Node<T> current = openList.Dequeue();
    Node<T> best = current;
    PathArrange(stack, current);
    current.IsVisited = true;

    do
    {
        foreach (var temp in current.Neighbors)...
        current.IsVisited = true;
        Node<T> Si = openList.Dequeue();
        if (Si.F <= current.F)
        {
            current = Si;
            PathArrange(stack, current);
            current.IsVisited = true;
        }
        else if (Si.F <= (current.F + temperature - current.Depth))
        {
            current = Si;
            PathArrange(stack, current);
            current.IsVisited = true;
        }
        temperature = temperature - 1;

    } while (openList.Any());
    return StackToList(stack);
}
```

İn this example, I have set the cooling as 1 so it decreases one by one.

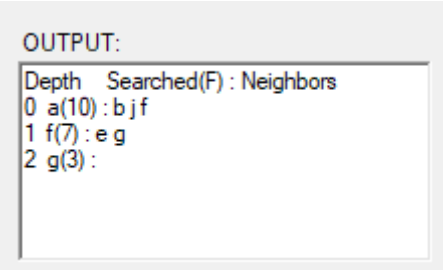
On the paper, I couldn't solve the given problem but I used the algorithm on a search tree. It is;



I created the tree and their heuristics.

```
private void Go()
{
    Node<string> a = new Node<string>("a", 10);
    Node<string> b = new Node<string>("b", 10);
    Node<string> j = new Node<string>("j", 8);
    Node<string> f = new Node<string>("f", 7);
    Node<string> e = new Node<string>("e", 5);
    Node<string> g = new Node<string>("g", 3);
    Node<string> k = new Node<string>("k", 0);
    Node<string> i = new Node<string>("i", 6);


    Graph<string> graph=new Graph<string>();

    graph.Add(a, b, 0);
    graph.Add(a, j, 0);
    graph.Add(a, f, 0);
    graph.Add(j, k, 0);
    graph.Add(f, e, 0);
    graph.Add(f, g, 0);
    graph.Add(e, i, 0);
    graph.Add(i, k, 0);


    Func<Node<string>, int> func = delegate (Node<string> item) { return item.F; };
    PriorityQueue<Node<string>, int> priorityQueue = new PriorityQueue<Node<string>, int>(func);
    priorityQueue.Enqueue(a);


    Searches<string> searches = new Searches<string>();
    List<Node<string>> list = searches.SimulatedAnnealing(priorityQueue, 5);
    TextBoxWriter(list);
    graph.UnvisitAll();
}
```

When I set the temperature 5, the result is;



When I set the temperature 6, the result is;