Laboratory work # 7

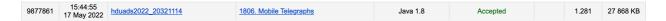
Student: HU Riqian Student ID: 20321114

Timus Name: hduads2022_20321114

Mail: jhlxhrq@163.com

Problem # 1806. Mobile Telegraph

Screenshot from Timus:



Explanation of algorithm:

- 1. Use the CMP model to sort the queue.
- 2. Use the DFS algorithm to do the searching.

Computational complexity of algorithm:

O(V)

Source code:

```
import java.io.*;
import java.util.*;
public class MobileTelegraph {
    static int SIZE = 50005, c = 0, n;
static boolean[] isConv = new boolean[SIZE];
    static boolean f = false;
    static int[] a = new int[SIZE];
    static long[] V = new long[SIZE];
    static Map<String, Integer> hm = new HashMap<>();
    static Comparator<Integer> cmp = (o1, o2) -> (int) (V[o1] - V[o2] != 0?
    static Queue<Integer> Q = new PriorityQueue<>(cmp);
    static String[] S = new String[SIZE];
    static T[] t = new T[SIZE];
    public static void main(String[] args) {
        new MobileTelegraph().run();
    static Scanner sc;
    static PrintWriter out;
    static StringBuilder sb;
```

```
void run() {
    sc = new Scanner(System.in);
    out = new PrintWriter(new OutputStreamWriter(System.out));
    sb = new StringBuilder();
void s() {
    int[] tg = new int[10];
    int now = 0;
    char c;
        tg[i] = sc.nextInt();
    for (int i = 0; i < n; ++i) {
        t[i] = new T(i);
    isConv[0] = true;
    a[MobileTelegraph.c] = 1;
    MobileTelegraph.c++;
    for (int i = 1; i < n || now != n - 1; ++i) {</pre>
                if ((char) (k + 48) != S[now].charAt(j)) {
                     sb.setCharAt(j, (char) (k + 48));
                     if (hm.get(sb.toString()) != null
                         st(tg, now);
        for (int j = 0; j < 10; ++j)</pre>
                if (hm.get(sb.toString()) != null
                    st(tg, now);
        if (now == n - 1) break;
        if (Q.peek() == null) break;
        isConv[now] = true;
void st(int[] tg, int now) {
```

```
int temp = hm.get(sb.toString()) - 1;
    if (V[\text{temp}] > V[\text{now}] + \text{tg}[v(S[\text{now}], sb.\text{toString}())]) {
         V[\text{temp}] = V[\text{now}] + \text{tg}[v(S[\text{now}], sb.\text{toString}())];
         Q.add(temp);
         t[now].add(temp, tg[v(S[now], sb.toString())]);
void pr() {
    } else {
         for (int i = 0; i < c; ++i) {
    while (true) {
         } else {
             break;
    return n;
boolean dfs(int n, int end) {
    boolean back = false;
    if (n == end) {
         f = true;
         back = true;
    for (P r : t[n].1) {
         if (f) break;
             continue;
         return false;
    } else {
         return true;
static class P {
    int w;
         this.w = w;
         this.to = t[1];
```

```
static class T {
    int r;
    ArrayList<P> l = new ArrayList<>();

    T(int r) {
        this.r = r;
    }

    void add(int l, int w) {
        this.l.add(new P(l, w));
    }
}
```