PAT甲级 1131 Subway Map

https://blog.csdn.net/m0_50617544/article/details/128523655

九是否随机的称呼 于 2023-02-25 18:07:02 发布

堆优化

```
#include<iostream>
    #include<vector>
   #include<algorithm>
    #include<queue>
   #include<unordered map>
   using namespace std;
   typedef pair<int, int> p;
    p r0, r1, r2;
    vector<int> pth[10006], tmp, res;
    vector v[10006], lne[10006];
   unordered map<unsigned int, unsigned int> ump;
11
    int inf = 999999999, status[10006];
   int tat, ending, minmin = inf, transfnum;
13
    int get transfnum() {
14
        int num = 0, pre = 0;
15
        for(int i = 1; i < tmp.size(); i++) {</pre>
16
            if(ump[tmp[i] * 10000 + tmp[i-1]] != pre) num++;
17
            pre = ump[tmp[i] * 10000 + tmp[i-1]];
18
19
20
        return num;
21
    void recursion(int edi) {
22
23
        tmp.push back(edi);
24
        if(tat==edi) {
            transfnum = get transfnum();
25
            if(transfnum < minmin){</pre>
26
                minmin = transfnum;
27
28
                res = tmp;
29
```

```
30
            tmp.pop_back();
                                         return;
32
33
        vector<int> kk = pth[edi];
34
        for(int i = 0; i < pth[edi].size(); i++) {</pre>
35
            recursion(pth[edi][i]);
36
37
        tmp.pop_back();
38
    int main(void) {
39
40
        int i, j, k, m, n, N, M, T, pre, ditance[10006], x, y, z;
41
        cin>>N;
        for(i = 1; i <= N; i++) {
42
43
            cin>>T;
            for(j = 0; j < T; j++) {
44
                cin>>m;
45
                if(j!=0) {
46
47
                    r0.first = 1;
48
                    r0.second = m;
                    v[pre].push_back(r0);
49
                    r0.second = pre;
50
                    v[m].push_back(r0);
51
52
                    ump[pre * 10000 + m] = ump[m * 10000 + pre] = i;
53
54
55
                pre = m;
56
57
        cin>>M;
58
        for(i = 0; i < M; i++) {
59
60
            cin>>tat>>ending;
61
            fill(status, status + 10006, 0);
            fill(ditance, ditance + 10006, inf);
62
            ditance[tat] = 0;
63
            priority_queue<p, vector<p>, greater> pq;
64
            r0.first = 0;
65
            r0.second = tat;
66
            pq.push(r0);
67
68
            while(!pq.empty()) {
```

```
r0 = pq.top(); 70
69
                                                 pq.pop();
71
                 x = r0.second;
72
                 if(status[x]==1) continue;
73
                 status[x] = 1;
                 for(j = 0; j < v[x].size(); j++) {</pre>
74
                     y = v[x][j].second;
75
                     k = v[x][j].first;
76
                     if(status[y]==0 && ditance[y] > ditance[x] + k) {
77
                         ditance[y] = ditance[x] + k;
78
79
                         pq.push({ditance[y], y});
80
                         pth[y].clear();
                         pth[y].push back(x);
81
                     } else if(status[y]==0 && ditance[y] == ditance[x] + k) {
82
                         pth[y].push_back(x);
83
84
85
86
             tmp.clear();
87
             minmin = inf;
88
             fill(status, status + 10006, 0);
89
             recursion(ending);
90
             int preline=0, nowline=9;
91
92
             vector<int> tp, lk;
93
             reverse(res.begin(), res.end());
             for(j = 1; j < res.size(); j++) {</pre>
94
                 nowline = ump[res[j] * 10000 + res[j - 1]];
95
                 if(preline!=nowline) {
96
                     tp.push_back(res[j - 1]);
97
                     lk.push_back(nowline);
98
99
100
                 preline = nowline;
101
             tp.insert(tp.end(), res[res.size() - 1]);
102
             lk.insert(lk.end(), nowline);
103
             printf("%d\n", res.size() - 1);
104
             for(j = 0; j < tp.size() - 1; j++) {
105
                 printf("Take Line#%d from %04d to %04d.\n",
106
107
                         lk[j], tp[j], tp[j+1]);
```

內容来源:csdn.net F者昵称:九是否随机的称呼 頁文链接:https://blog.csdn.net/m0_50617544/article/details/1285236 F者主页:https://blog.csdn.net/m0_50617544

```
108
                }<sub>109</sub>
           return 0;
110
111 }
```

提交结果

•	1
	(
	1

提交时间	状态 ①	分数	题目	编译器	内存	用时	用户
2023/01/23 09:31:23	答案正确	S ^{OO} 30	1131	C++ (g++)	2620 KB	17 ms	
测试点	结果		分数		耗时		内存
0	答案正确		15		4 ms		1080 KB
1	答案正确		3		4 ms		1080 KB
2	答案正确		3		4 ms		1212 KB
3	答案正确		5		4 ms		1208 KB
4	答案正确		1		4 ms		1096 KB
5	答案正确		3		17 ms	CSDN @力	2620 KB 是否随机的称呼

update2

```
2 #include<vector>
                      3 #include<unordered map>
    #include<algorithm>
 5
    using namespace std;
    unordered_map<int, int> ump;
6
    vector<int> v[10006], tmp, res;
   int inf = 999999999, status[10006];
    int starts, ending, transfnum, numnode, shift = 10000;
    int gettransf() {
10
        int preline = 0, num = 0, tpl;
11
12
        for(int i = 1; i < tmp.size(); i++) {</pre>
            tpl = ump[tmp[i] * shift + tmp[i-1]];
13
            if(preline!=tpl) num++;
14
15
            preline = tpl;
16
17
        return num;
18
19
    void recursion(int st) {
        if(st == ending && tmp.size() <= numnode) {</pre>
20
            int num_transf = gettransf();
21
            if(tmp.size() < numnode ||</pre>
22
                (tmp.size()==numnode && num_transf < transfnum)) {</pre>
23
                transfnum = num transf;
24
                numnode = tmp.size();
25
26
                res = tmp;
27
28
            return;
29
        for(int i = 0; i < v[st].size(); i++) {</pre>
30
            int pt = v[st][i]; // local variable
31
            if(status[pt]==1) continue;
32
33
            status[pt] = 1;
            tmp.push_back(pt);
34
            recursion(pt);
35
            tmp.pop_back();
36
37
            status[pt] = 0;
38
39
40
   int main(void) {
```

内容来源:csdn.net 作者昵称:九是否随机的称呼 原文链接:https://blog.csdn.net/m0_50617544/article/details/1285236 作者主页:https://blog.csdn.net/m0_50617544

```
int i, j, k, m, n, y, z, pre;
42
41
                                               cin>>m;
        for(i = 1; i <= m; i++) {</pre>
43
            cin>>n;
44
45
            for(j=0; j < n; j++) {
                cin>>y;
46
47
                if(j > 0) {
                    ump[pre * shift + y] = ump[y * shift + pre] = i;
48
                    v[y].push_back(pre);
49
                    v[pre].push_back(y);
50
51
52
                pre = y;
53
54
55
        cin>>n;
        for(i = 0; i < n; i++) {
56
57
            cin>>starts>>ending;
58
            tmp.clear();
            tmp.push_back(starts);
59
            transfnum = numnode = inf;
60
            fill(status, status + 10006, 0);
61
            recursion(starts);
62
            vector<int> vrnode, vrline;
63
            int preline = -1, nowline = -2;
64
65
            for(j = 1; j < res.size(); j++) {</pre>
                nowline = ump[res[j] * shift + res[j - 1]];
66
                if(nowline!=preline) {
67
                    vrnode.push_back(res[j - 1]);
68
                    vrline.push back(nowline);
69
70
71
                preline = nowline;
72
            vrnode.push back(res[res.size() - 1]);
73
            vrline.push_back(nowline);
74
            printf("%d\n", res.size() - 1);
75
            for(j = 0; j < vrnode.size() - 1; j++) {</pre>
76
77
                printf("Take Line#%d from %04d to %04d.\n",
78
                    vrline[j], vrnode[j], vrnode[j+1]);
79
```

提交结果

- 2	K
	7

提交时间	状态①	分数	题目	编译器	内存		用时	用户
2023/01/23 09:32:17	答案正确	30	1131	C++ (g++)	1720	KB	41 ms	
测试点	结果		分数			耗时		内存
0	答案正确		15			3 ms		584 KB
CP ^{ON}	答案正确		350			4 ms		696 KB
2	答案正确		3			4 ms		700 KB
3	答案正确		5			41 ms		572 KB
4	答案正确		150			4 ms		696 KB
5	答案正确		3			12 ms	CSDN @九	1720 KB 是否随机的称呼

update20230225 recursion

1 #include<iostream>

2 #include<vector>

内容来源: csdn net

作者昵称・九是否随机的称

压力性较,144、144、144、0.50045

原文链接: https://blog.csdn.net/m0 50617544/article/details/1285/

作者主页: https://blog.csdn.net/m0 5061754

```
3 #include<unordered_map>
                            4 #include<algorithm>
   using namespace std;
   unordered_map<int, int> ump;
   vector<int> res, tmp, route[10006];
   int status[10006], start, endkk, mintra, mindik;
 8
    int gettransfer() {
9
        int num = 0, pretran = 0;
10
        for(int i = 0; i < tmp.size(); i++) {</pre>
11
            if(i > 0) {
12
13
                if(pretran!=ump[tmp[i-1]*10000 + tmp[i]]) num++;
                pretran = ump[tmp[i-1]*10000 + tmp[i]];
14
15
16
17
        return num;
18
    void recursion(int station) {
19
20
        if(station == endkk) {
21
            int num = gettransfer();
            if(tmp.size() < mindik || (tmp.size()==mindik && num < mintra)) {</pre>
22
                mindik = tmp.size();
23
24
                mintra = num;
25
                res = tmp;
26
27
            return;
28
        for(int i = 0; i < route[station].size(); i++) {</pre>
29
            int kk = route[station][i];
30
            if(status[kk] == 0) {
31
                status[kk] = 1;
32
                tmp.push_back(kk);
33
                recursion(kk);
34
                tmp.pop_back();
35
                status[kk] = 0;
36
37
38
39
40
    int main(void) {
41
        int i, j, k, m, n, N, M, K, x, y, z, cnt, pre;
```

```
42
        cin>>N;
                       for(i = 1; i <= N; i++) {
               43
44
            cin>>M;
45
            for(j = 1; j <= M; j++) {
                cin>>y;
46
47
                if(j > 1) {
                    ump[pre * 10000 + y] = ump[y * 10000 + pre] = i;
48
                    route[pre].push_back(y);
49
                    route[y].push_back(pre);
50
51
52
                pre = y;
53
54
55
        cin>>K;
        for(i = 0; i < K; i++) {
56
57
            cin>>start>>endkk;
58
            tmp.clear();
59
            res.clear();
            tmp.push_back(start);
60
            mintra = mindik = 999999999;
61
            fill(status, status + 10006, 0);
62
            status[start] = 1;
63
            recursion(start);
64
            printf("%d\n", res.size() - 1);
65
            int pretra = ump[res[0] * 10000 + res[1]], now = res[0];
66
            for(j = 1; j < res.size(); j++) {</pre>
67
                if(pretra != ump[res[j - 1] * 10000 + res[j]]) {
68
                    printf("Take Line#%d from %04d to %04d.\n", pretra, now, res[j-1]);
69
70
                    now = res[j - 1];
71
72
                pretra = ump[res[j-1]*10000 + res[j]];
73
            printf("Take Line#%d from %04d to %04d.\n", pretra, now, res[j-1]);
74
75
76
        return 0;
77 }
```

内容来源: csdn ne

作者昵称: 九是否随机的称呼

原文链接: https://blog.csdn.net/m0_50617544/article/details/1285230

作类主页: https://blog.codp.not/m0_506175

```
#include<iostream>
   #include<queue>
    #include<vector>
   #include<algorithm>
    #include<unordered map>
   using namespace std;
   typedef pair<int, int> p;
   int status[10006], mintra, start, endkk, dik[10006], inf = 9999999999;
   vector<int> route[10006], tmp, res;
   unordered map<int, int> ump;
   int gettranfer() {
11
12
       int num = 0, pretra = 0;
       for(int i = 0; i < tmp.size(); i++) {</pre>
13
14
            if(i > 0) {
                if(pretra != ump[tmp[i-1] * 10000 + tmp[i]]) num++;
15
16
            pretra = ump[tmp[i-1] * 10000 + tmp[i]];
17
18
19
        return num;
20
    void recursion(int station) {
21
        tmp.push_back(station);
22
23
       if(station==start) {
            int num = gettranfer();
24
25
            if(mintra > num) {
26
                res = tmp;
27
                mintra = num;
28
            tmp.pop_back();
29
30
            return;
31
        for(int i = 0; i < route[station].size(); i++) {</pre>
32
            recursion(route[station][i]);
33
34
        tmp.pop_back();
35
36
37
    int main(void) {
38
        int i, j, k, m, n, N, M, K, x, y, z, pre;
```

```
priority_queue<p, vector<p>, greater> pq;
40
39
                                                             p p0, p1, p2;
41
        vector v[10006];
42
        cin>>N;
        for(i = 1; i <= N; i++) {
43
44
            cin>>M;
            for(j = 0; j < M; j++) {
45
46
                cin>>y;
47
                if(j > 0) {
48
                    v[y].push_back({1, pre});
                    v[pre].push_back({1, y});
49
                    ump[pre * 10000 + y] = ump[y * 10000 + pre] = i;
50
51
52
                pre = y;
53
54
        cin>>K;
55
        for(i = 0; i < K; i++) {
56
57
            cin>>start>>endkk;
            fill(status, status + 10006, 0);
58
59
           fill(dik, dik + 10006, inf);
60
            mintra = inf;
            dik[start] = 0;
61
            pq.push({0, start});
62
            while(!pq.empty()) {
63
                p0 = pq.top();
64
                x = p0.second;
65
                pq.pop();
66
                if(status[x] == 1) continue;
67
                status[x] = 1;
68
                for(j = 0; j < v[x].size(); j++) {</pre>
69
70
                    p1 = v[x][j];
71
                    y = p1.second;
72
                    z = p1.first;
                    if(status[y] == 0 \&\& dik[y] > dik[x] + z) {
73
74
                        dik[y] = dik[x] + z;
75
                        pq.push({dik[y], y});
76
                        route[y].clear();
77
                        route[y].push_back(x);
```

```
} else if(status[y] == 0 && dik[y] == dik[x] + z) {79
78
                     route[y].push_back(x);
81
                }
82
83
            tmp.clear();
            recursion(endkk);
84
            reverse(res.begin(), res.end());
85
            printf("%d\n", res.size() - 1);
86
            int pretra = ump[res[0] * 10000 + res[1]], now=res[0];
87
            for(j = 1; j < res.size(); j++) {</pre>
88
89
                if(pretra!=ump[res[j]*10000 + res[j-1]]) {
90
                    printf("Take Line#%d from %04d to %04d.\n", pretra, now, res[j-1]);
91
                    now = res[j-1];
92
                pretra = ump[res[j]*10000 + res[j-1]];
93
94
            printf("Take Line#%d from %04d to %04d.\n", pretra, now, res[j-1]);
95
96
97
        return 0;
98 }
```

update

```
#include<iostream>
    #include<vector>
    #include<unordered_map>
    #include<algorithm>
   using namespace std;
 5
   unordered_map<int, int> ump;
    vector<int> v[10006], tmp, res;
   int starts, endst, minmin = 9999999999, quick = 999999999;
    int status[10006];
9
   int gettransf() {
10
11
        int preline = 0, z, cnt = 0;
        for(int i = 1; i < tmp.size(); i++) {</pre>
12
            z = ump[tmp[i-1]*10000 + tmp[i]];
13
14
            if(preline!=z) cnt++;
```

```
preline = z;_{16}
15
17
        return cnt;
18
    void recursion(int st) {
19
        if(endst==st && tmp.size() <= quick) {</pre>
20
21
            int num_transf = gettransf();
22
            if(tmp.size() < quick||</pre>
23
            (tmp.size() == quick && minmin > num_transf)) {
24
                res = tmp;
25
                minmin = num transf;
                quick = tmp.size();
26
27
28
            return;
29
        status[st]=1;
30
        vector<int> small = v[st];
31
        for(int i = 0; i < small.size(); i++) {</pre>
32
33
            if(status[small[i]]==1) continue;
            tmp.push_back(small[i]);
34
35
            status[small[i]]=1;
36
            recursion(small[i]);
37
            tmp.pop_back();
            status[small[i]]=0;
38
39
40
    int main(void){
41
42
        int i, j, k, m, n, y, z, pre;
43
        cin>>m;
        for(i = 0; i < m; i++) {
44
            cin>>n;
45
            for(j=0; j < n; j++) {
46
47
                cin>>y;
                if(j > 0) {
48
49
                    ump[pre * 10000 + y] = ump[y*10000 + pre] = i+1;
50
                    v[pre].push_back(y);
                    v[y].push_back(pre);
51
52
53
                pre = y;
54
```

```
55
56
        cin>>n;
        for(i = 0; i < n; i++) {
57
58
            cin>>starts>>endst;
           fill(status, status+10006, 0);
59
60
            tmp.clear();
61
            minmin = 9999999999;
62
            tmp.push_back(starts);
63
            recursion(starts);
64
           printf("%d\n", res.size() - 1);
65
           int preline = 0, pretransf = starts;
66
           for(j = 1; j < res.size(); j++) {</pre>
67
                z = ump[res[j-1]*10000+res[j]];
68
               if(z!=preline){
69
                   if(preline!=0) printf("Take Line#%d from %04d to %04d.\n",
70
                                           preline, pretransf, res[j-1]);
71
                   preline = z;
72
                   pretransf = res[j-1];
73
74
75
            printf("Take Line#%d from %04d to %04d.\n", preline, pretransf, res[j-1]);
76
77
       return 0;
78
79 }
```

old before

```
#include<iostream>
#include<vector>
#include<unordered_map>
#include<algorithm>
using namespace std;
int minmin, inf = 30000, query[11][2], start, endes, num;
bool status[9000];
unsigned char costes[9000];
unsigned char costes[9000];
unordered_map<int, int> ump, rev_ump, subway, all;

#include<iostream>
#include<io
```

```
vector<vector<int>> route;11 | vector<int> pth, tmppth;
12
    void recursion(int station) {
13
        tmppth.push_back(rev_ump[station]);
14
        if(station==ump[start]) {
15
            int pre = 0, cmn = 0;
16
            for(int j = 1; j < tmppth.size(); j++) {</pre>
                int sub = subway[ump[tmppth[j]]*10000+ump[tmppth[j-1]]];
17
                if(sub!=pre){
18
19
                    pre = sub;
20
                    cmn++;
21
22
            if(cmn < minmin) {</pre>
23
                minmin = cmn;
24
25
                pth = tmppth;
26
27
28
        for(int i = 0; i < route[station].size(); i++) {</pre>
            recursion(route[station][i]);
29
30
        tmppth.pop_back();
31
32
33
    int main(int argc, char **argv) {
34
        int i, j, k, m, n, y, z, cnt = 0, pre, begin, tailes;
35
        cin>>m;
        for(i = 0; i < m; i++) {
36
37
            cin>>n;
            for(j = 0; j < n; j++) {
38
39
                cin>>y;
40
                if(ump[y]==0) {
41
                    rev_ump[cnt] = y;
                    ump[y] = cnt++;
42
43
                if(j >= 1) {
44
                    all[pre*10000+ump[y]] = all[ump[y]*10000+pre] = 1;
45
                    subway[pre*10000+ump[y]]=subway[ump[y]*10000+pre] = i+1;
46
47
                pre = ump[y];
48
49
```

```
50
         51
                 cin>>m;
        for(k = 0; k < m; k++) {
52
53
            cin>>start>>endes;
54
            fill(status, status+9000, false);
            fill(costes, costes+9000, inf);
55
56
            costes[ump[start]] = 0;
57
            route.clear();
58
            route.resize(9000);
            for(i = 0; i < cnt; i++) {</pre>
59
                minmin = inf;
60
                for(j = 0; j < cnt; j++) {
61
                    if(status[j]==false&&minmin > costes[j]) {
62
                       minmin = costes[j];
63
64
                        y = j;
65
66
                status[y] = true;
67
68
                for(j = 0; j < cnt; j++) {
                   if(status[j]==false&&all[y*10000+j]!=0&&costes[j] > costes[y] + 1) {
69
70
                        costes[j] = costes[y] + 1;
71
                       route[j].clear();
72
                        route[j].push_back(y);
                    s=10000+j!=0&&costes[j] == costes[y] + 1) {
73
                       route[j].push_back(y);
74
75
76
77
78
            minmin = inf;
            tmppth.clear();
79
80
            pth.clear();
81
            recursion(ump[endes]);
82
            reverse(pth.begin(), pth.end());
83
            printf("%d\n", pth.size() - 1);
84
            int pre = 0, pretransfer = start;
85
            for(int j = 1; j < pth.size(); j++) {</pre>
               if(subway[ump[pth[j-1]]*10000+ump[pth[j]]]!=pre) {
86
                   if(pre!=0) printf("Take Line#%d from %04d to %04d.\n", pre, pretransfer, pth[j-1]);
87
88
                    pre = subway[ump[pth[j-1]]*10000+ump[pth[j]]];
```

GitHub - ZouJiu1/PAT: 浙江大学PAT题目解答内容

浙江大学PAT题目解答内容. Contribute to ZouJiu1/PAT development by creating an account on GitHub.

https://github.com/ZouJiu1/PAT

内容来源: csdn.net 作者昵称: 九是否随机的称呼 原文链接: https://blog.csdn.net/m0_50617544/article/details/12852365 作者主页: https://blog.csdn.net/m0_50617544