

原创 树状DP入门例题

2019-08-22 13:30:57 _Y-_Y-_ 阅读数 66 更多

编辑

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/weixin_44410512/article/details/100009738

P1352 没有上司的舞会

```
1  /*
2  没有上司的舞会
3  P1352
4  https://www.luogu.org/problem/P1352
5  题意：有一场舞会，每一个人都有一个快乐值，但不能和上司同时出现在舞会中，问这个舞会的快乐值最大为多少
6  解法：
7  树状DP
8  */
9  #include <bits/stdc++.h>
10 using namespace std;
11 #define maxn 100000
12 struct node{
13     int u,v;
14     int next;
15 }edge[maxn];
16 int head[maxn],cnt;
17 int dp[maxn][2];
18 int a[maxn];
19 void init(){
20     memset(head,-1,sizeof head);
21     cnt=0;
22 }
23 void add(int u,int v){
24     edge[cnt].u=u;
25     edge[cnt].v=v;
26     edge[cnt].next=head[u];
27     head[u]=cnt++;
28 }
29 void dfs(int u,int fa){
30     dp[u][0]=0;
31     dp[u][1]=a[u];
32     for(int i=head[u];~i;i=edge[i].next){
33         int v=edge[i].v;
34         if(v!=fa){
35             dfs(v, u);
36             dp[u][0]+=max(dp[v][0],dp[v][1]);
37             dp[u][1]+=dp[v][0];
38         }
39     }
40 }
41 int main(){
42     int n,u,v;
43     cin>>n;
44     init();
45     for(int i=1;i<=n;i++) cin>>a[i];
46     while(cin>>u>>v&&u+v){
47         add(u,v);
48         add(v,u);
49     }
50     dfs(1,-1);
51     cout<<max(dp[1][0],dp[1][1])<<endl;
52     return 0;
53 }
54
```

The more, The Better

编程语言大PK，你选谁？

关闭

```

1  /*
2  The more, The Better
3  HDU - 1561
4  https://cn.vjudge.net/problem/HDU-1561
5  题意: 每一个城市都有一定的宝物, 允许攻克m个城堡, 有些城堡不能直接攻克, 要攻克它必须先攻克指定的城堡, 问最多可以得到多少宝物
6  解法:
7  树状DP+01背包
8  */
9  #include <cstdio>
10 #include <cstring>
11 #include <iostream>
12 using namespace std;
13 #define maxn 1000
14 int dp[maxn][maxn];
15 int f[maxn][maxn];
16 int as[maxn];
17 struct node{
18     int u,v;
19     int next;
20 }edge[maxn];
21 int head[maxn];
22 int cnt;
23 int n,m,a,b;
24 void init(){
25     memset(head,-1,sizeof head);
26     memset(dp,0,sizeof dp);
27     memset(f,0,sizeof f);
28     memset(as,0,sizeof as);
29
30     cnt=0;
31 }
32 void add(int u,int v){
33     edge[cnt].u=u;
34     edge[cnt].v=v;
35     edge[cnt].next=head[u];
36     head[u]=cnt++;
37 }
38 void dfs(int u){
39     for(int i=head[u];~i;i=edge[i].next){
40         int v=edge[i].v;
41         dfs(v);
42         for(int k=m;k>=0;k--){
43             for(int j=0;j<=k;j++){
44                 f[u][k]=max(f[u][k],f[u][k-j]+dp[v][j]);
45             }
46         }
47     }
48     for(int i=1;i<=m+1;i++){
49         dp[u][i]=f[u][i-1]+as[u];
50     }
51 }
52 int main(){
53     while(cin>>n>>m&&n+m){
54         init();
55         for(int i=1;i<=n;i++){
56             cin>>a>>b;
57             as[i]=b;
58             add(a,i);
59         }
60         dfs(0);
61         printf("%d\n",dp[0][m+1]);
62     }
63     return 0;
64 }

```

Binary Apple Tree

解法一:

编程语言大PK，你选谁？

关闭

```

1  /*
2  Binary Apple Tree
3  URAL - 1018
4  https://cn.vjudge.net/problem/URAL-1018
5  题意：有一棵二叉苹果树，每个节点都有苹果，现在让你保留下来m条枝头，问最多可以剩余多少个苹果
6  解法：
7  树状DP + 01背包
8  */
9  #include <cstdio>
10 #include <cstring>
11 #include <iostream>
12 using namespace std;
13 #define maxn 1000
14 int f[maxn][maxn];
15 int as[maxn];
16 struct node{
17     int u,v,w;
18     int next;
19 }edge[maxn];
20 int head[maxn];
21 int cnt;
22 int n,m;
23 void init(){
24     memset(head,-1,sizeof head);
25     memset(f,0,sizeof f);
26     cnt=0;
27 }
28 int max(int a,int b){
29     if(a>b) return a;
30     return b;
31 }
32 void add(int u,int v,int w){
33     edge[cnt].u=u;
34     edge[cnt].v=v;
35     edge[cnt].w=w;
36     edge[cnt].next=head[u];
37     head[u]=cnt++;
38 }
39 void dfs(int u,int fa){
40     for(int i=head[u];~i;i=edge[i].next){
41         int v=edge[i].v;
42         if(v!=fa){
43             dfs(v,u);
44             for(int k=m;k>=0;k--){
45                 for(int j=1;j<=k;j++){
46                     if(f[u][k]<f[u][k-j]+f[v][j-1]+edge[i].w)
47                         f[u][k]=f[u][k-j]+f[v][j-1]+edge[i].w;
48                 }
49             }
50         }
51     }
52 }
53 int main(){
54     int a,b,c;
55     while(cin>>n>>m){
56         init();
57         for(int i=1;i<n;i++){
58             cin>>a>>b>>c;
59             add(a,b,c);
60             add(b,a,c);
61         }
62         dfs(1,1);
63         printf("%d\n",f[1][m]);
64     }
65     return 0;
66 }
67

```

解法二：

```

1  /*
2  Binary Apple Tree
3  URAL - 1018
4  https://cn.vjudge.net/problem/URAL-1018
5  题意：有一棵二叉苹果树，每个节点都有苹果，现在让你保留下来m条枝头，问最多可以剩余多少个苹果
6  解法：
7  dfs + 记忆化 + 枚举
8  */
9  #include <cstdio>
10 #include <cstring>
11 #include <iostream>
12 using namespace std;
13 #define maxn 1000
14 int dp[maxn][maxn];
15 int f[maxn][maxn];
16 int as[maxn];
17 struct node{
18     int u,v,w;
19     int next;
20 }edge[maxn];
21 int head[maxn];
22 int cnt;
23 int n,m;
24 void init(){
25     memset(head,-1,sizeof head);
26     memset(dp,0,sizeof dp);
27     cnt=0;
28 }
29 int max(int a,int b){
30     if(a>b) return a;
31     return b;
32 }
33 void add(int u,int v,int w){
34     edge[cnt].u=u;
35     edge[cnt].v=v;
36     edge[cnt].w=w;
37     edge[cnt].next=head[u];
38     head[u]=cnt++;
39 }
40 void dfs(int u,int fa){
41     for(int i=head[u];~i;i=edge[i].next){
42         int v=edge[i].v;
43         if(v!=fa){
44             as[v]=edge[i].w;
45             dfs(v,u);
46             for(int k=m;k>=0;k--){
47                 for(int j=0;j<=k;j++){
48                     f[u][k]=max(f[u][k],f[u][k-j]+dp[v][j]);
49                 }
50             }
51         }
52     }
53     for(int i=1;i<=m+1;i++){
54         dp[u][i]=f[u][i-1]+as[u];
55     }
56 }
57 int main(){
58     int a,b,c;
59     while(cin>>n>>m){
60         init();
61         for(int i=1;i<n;i++){
62             cin>>a>>b>>c;
63             add(a,b,c);
64             add(b,a,c);
65         }
66         dfs(1,1);
67         printf("%d\n",dp[1][m+1]);
68     }
69     return 0;
70 }

```

Strategic game

```

1  /*
2  Strategic game
3  POJ - 1463
4  https://cn.vjudge.net/problem/POJ-1463
5  题面: 最小点覆盖
6  解法: 树状DP
7  */
8  #include <iostream>
9  #include <cstring>
10 #include <algorithm>
11 #include <cstdio>
12 using namespace std;
13 #define maxn 100000
14 struct node{
15     int u,v;
16     int next;
17 }edge[maxn];
18 int head[maxn];
19 int dp[maxn][2];
20 int cnt;
21 void init(){
22     memset(head,-1,sizeof head);
23     memset(dp,0,sizeof dp);
24     cnt=0;
25 }
26 void add(int u,int v){
27     edge[cnt].u=u;
28     edge[cnt].v=v;
29     edge[cnt].next=head[u];
30     head[u]=cnt++;
31 }
32 void dfs(int u,int fa){
33     //cout<<u<<endl;
34     dp[u][0]=0;
35     dp[u][1]=1;
36     for(int i=head[u];~i;i=edge[i].next){
37         int v=edge[i].v;
38         if(v!=fa){
39             dfs(v,u);
40             dp[u][0]+=dp[v][1];
41             dp[u][1]+=min(dp[v][0],dp[v][1]);
42         }
43     }
44 }
45 int main(){
46     int n,m,k,a;
47     while(cin>>n){
48         init();
49         for(int i=0;i<n;i++){
50             scanf("%d:%d",&m,&k);
51             for(int i=0;i<k;i++){
52                 cin>>a;
53                 add(m,a);
54                 add(a,m);
55             }
56         }
57         dfs(0,0);
58         cout<<min(dp[0][0],dp[0][1])<<endl;
59     }
60     return 0;
61 }
62

```

Cell Phone Network

```

1  /*
2  Cell Phone Network
3

```

```

4 POJ - 3659
5 https://cn.vjudge.net/problem/POJ-3659
6 题面: 最小支配集
7 解法: 树状DP
8 */
9 #include <iostream>
10 #include <cstring>
11 #include <algorithm>
12 #include <cstdio>
13 using namespace std;
14 #define maxn 500000
15 #define INF 0x3f3f3f3f
16 struct node{
17     int u,v;
18     int next;
19 }edge[maxn];
20 int head[maxn];
21 int dp[maxn][3];
22 int cnt;
23 void init(){
24     memset(head,-1,sizeof head);
25     memset(dp,0,sizeof dp);
26     cnt=0;
27 }
28 void add(int u,int v){
29     edge[cnt].u=u;
30     edge[cnt].v=v;
31     edge[cnt].next=head[u];
32     head[u]=cnt++;
33 }
34 void dfs(int u,int fa){
35     dp[u][0]=1;
36     dp[u][1]=0;
37     dp[u][2]=0;
38     int sum=0,inc=INF;
39     bool judge=false;
40     for(int i=head[u];~i;i=edge[i].next){
41         int v=edge[i].v;
42         if(v!=fa){
43             dfs(v,u);
44             dp[u][0]+=min(dp[v][0],min(dp[v][1],dp[v][2]));
45             if(dp[v][0]<=dp[v][1]){
46                 sum+=dp[v][0];
47                 judge=true;
48             }else{
49                 sum+=dp[v][1];
50                 inc=min(inc,dp[v][0]-dp[v][1]);
51             }
52             if(dp[v][1]!=INF&&dp[u][2]!=INF) dp[u][2]+=dp[v][1];
53             else dp[u][2]=INF;
54             dp[u][1]+=min(dp[v][0],dp[v][1]);
55         }
56     }
57     if(inc==INF&&!judge) dp[u][1]=INF;
58     else {
59         dp[u][1]=sum;
60         if(!judge) dp[u][1]+=inc;
61     }
62 }
63 int main(){
64     int n,u,v;
65     while(cin>>n){
66         init();
67         for(int i=1;i<n;i++){
68             cin>>u>>v;
69             add(u,v);
70             add(v,u);
71         }
72         dfs(1,1);
73         cout<<min(dp[1][0],min(dp[1][1],dp[1][2]+1))<<endl;
74     }

```

```
75 | return 0;  
76 | }
```

有 0 个人打赏

文章最后发布于: 2019-08-22 13:30:57

编程语言大PK，你选谁？

关闭