f(i,j)=max{f(lc[i],k)+f(rc[i],j-k-1)}+date[i]; 0<=k<j;

f(i,j):=max(f(i.left,k)+f(i.right,j-k)),0<=k<=j

参考代码:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60 | #include<cstring>  #include<cstdio>  #include<iostream>  using namespace std;  struct node  {  int lc,rc;  int s;  } tree[220];  int F[110][110],apple[110][110];  bool vis[110];  int n,q;  void make\_tree(int root)  {  vis[root]=true;  for (int i=1;i<=n;i++)  if (!vis[i] && apple[root][i]!=-1)  {  if (tree[root].lc==0) tree[root].lc=i;  else tree[root].rc=i;  tree[i].s=apple[root][i];  make\_tree(i);  }  }  int tree\_dp(int t,int k)  {  if (F[t][k]!=-1) return F[t][k];  if (t==0 || k==0)  {  F[t][k]=0; return 0;  }  F[t][k]=0;  for (int i=0;i<=k-1;i++)  {  int ls=tree\_dp(tree[t].lc,i);  int rs=tree\_dp(tree[t].rc,k-1-i);  if (F[t][k]<ls+rs) F[t][k]=ls+rs;  }  F[t][k]+=tree[t].s;  return F[t][k];  }  int main()  {  scanf("%d%d",&n,&q); q++;  memset(apple,-1,sizeof(apple));  int a,b,s;  for (int i=1;i<=n-1;i++)  {  scanf("%d%d%d",&a,&b,&s);  apple[a][b]=s; apple[b][a]=s;  }  memset(tree,0,sizeof(tree));  memset(vis,false,sizeof(vis));  make\_tree(1);  memset(F,-1,sizeof(F));  int ans=tree\_dp(1,q);  printf("%d\n",ans);  return 0;  } |