#include <cstdio>

#include <cstring>

#include <algorithm>

#include <vector>

#define ll long long

#define MN 500005

#define MS 20

using namespace std;

struct edge{int nex,to;}e[MN];

int dfbg[MN],dfed[MN],dep[MN],hr[MN],q[MN],siz[MN],fa[MS][MN],b[MN],u[MN];

ll t[MN],ans[MN];

vector <int> d[MN];

int dfn,pin,bin,tp,n,rt;

inline int read()

{

int n=0,f=1; char c=getchar();

while (c<‘0‘ || c>‘9‘) {if(c==‘-‘)f=-1; c=getchar();}

while (c>=‘0‘ && c<=‘9‘) {n=n\*10+c-‘0‘; c=getchar();}

return n\*f;

}

inline void ins(int x,int y) {e[++pin]=(edge){hr[x],y}; hr[x]=pin;}

void dfs(int x,int depth)

{

dfbg[x]=++dfn; dep[x]=depth;

d[depth].push\_back(x);

for (register int i=hr[x];i;i=e[i].nex)

dfs(e[i].to,depth+1);

dfed[x]=dfn;

}

int lca(int x,int y)

{

register int i,k;

if (dep[x]<dep[y]) swap(x,y);

for (k=dep[x]-dep[y],i=0;k;k>>=1,++i)

if (k&1) x=fa[i][x];

if (x==y) return x;

for (i=MS-1;i>=0;--i)

if (fa[i][x]!=fa[i][y]) x=fa[i][x],y=fa[i][y];

return fa[0][x];

}

inline void pushs(int x) {while (tp&&dfbg[x]>dfed[q[tp]]) --tp; if (tp) ins(q[tp],x); q[++tp]=x;}

inline int lowbit(int x) {return x&-x;}

inline void getadd(int x,int z) {for (;x<=n;x+=lowbit(x)) t[x]+=z;}

inline ll getsum(int x) {ll lt=0; for (;x;x-=lowbit(x)) lt+=t[x]; return lt;}

void dp(int x,int fat)

{

siz[x]=u[x];

for (register int i=hr[x];i;i=e[i].nex)

dp(e[i].to,x),siz[x]+=siz[e[i].to];

getadd(dfbg[x]+1, 1LL\*siz[x]\*(dep[x]-dep[fat]));

getadd(dfed[x]+1,-1LL\*siz[x]\*(dep[x]-dep[fat]));

}

bool cmp(int x,int y) {return dfbg[x]<dfbg[y];}

int main()

{

register int i,j;

n=read(); bin=0;

for (i=1;i<=n;++i) ins(fa[0][i]=read(),i),b[++bin]=i;

for (i=1;i<=n;++i) if (!fa[0][i]) rt=i;

for (i=1;i<MS;++i)

for (j=1;j<=n;++j) fa[i][j]=fa[i-1][fa[i-1][j]];

dfs(rt,1);

for (i=2;d[i].size();++i)

{

for (j=1;j<=bin;++j) hr[b[j]]=u[b[j]]=0;

tp=pin=bin=0;

for (j=0;j<d[i].size();++j) ++u[b[++bin]=fa[0][d[i][j]]];

bin=unique(b+1,b+bin+1)-b-1;

for (j=1;j<bin;++j) b[bin+j]=lca(b[j],b[j+1]);

bin=bin\*2-1; sort(b+1,b+bin+1,cmp);

bin=unique(b+1,b+bin+1)-b-1;

for (j=1;j<=bin;++j) pushs(b[j]);

dp(q[1],0);

for (j=0;j<d[i].size();++j) ans[d[i][j]]=getsum(dfbg[d[i][j]]);

}

for (i=1;i<=n;++i) printf("%I64d ",ans[i]);

}