C++算法

快快编程1805

快快编程1804

思路A 简化问题

A 简化问题

图 | DAG | 树 / 环 | 链 | 1点 / 2点 / 3点

一路简化

极简问题优先解决

1点0边	答案是1
2点1边u->v	答案是0.5*2+0.5*1

链状问题

f[k]代表剩k点单向链所需次数期望

思路A 简化为链状

f[k]代表剩k点单向链所需次数期望

初始化	f[0]=0	f[1]=1

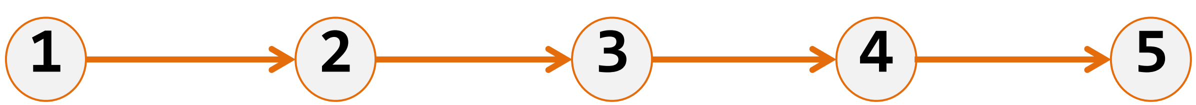
怎么解	
递推式	f[k]=1+1/k*(f[0]+f[1]+..+f[k-1])
差分抵消	f[k+1]=1+1/(k+1)*(f[0]+f[1]+..+f[k])
	f[k+1]*(k+1)-f[k]*k=1+f[k]
累加	f[k+1]-f[k]=1/(k+1)
	f[n]=1+1/2+1/3+..+1/n

总结结论,诠释公式,尝试推广

思路B 期望要素

B	期望要素	转换角度	求个体对期望的贡献

链状问题	求$E[X]$，其中随机变量X的含义为这次随机试验里删了几次才删完
单点贡献	拆解 $X=I_1+I_2+...+I_n$
	$E[X]=E[I_1]+E[I_2]+...+E[I_n]$
	其中随机01变量I_u的含义为这次随机试验里是否选中u号删除

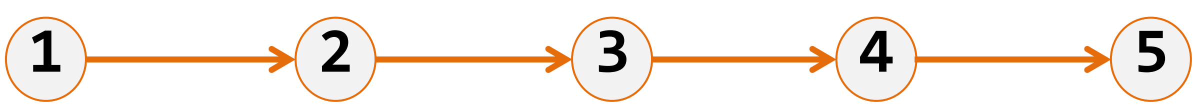


思路B 期望要素

$E[I_1]=1$	代表在所有随机试验中1号点被选中直接删除的次数期望
$E[I_2]=1/2$	代表在所有随机试验中2号点被选中直接删除的次数期望

选2号前,1,2一定都在,两者等概率被选中

$E[I_k]=1/k$	选k号前,k所有前驱也都在,都等概率



$E[I_1]=1$	$E[I_2]=1/2$	$E[I_1]=1/3$	$E[I_1]=1/4$	$E[I_1]=1/5$

思路B 期望要素

I_u是个随机01变量,其含义为
这次随机试验里是否选中u号删除

$E[I_u]$代表在所有随机试验中u号点
被选中直接删除的次数期望

$E[I_k]=1/c[k]$

c[k]代表图里有几个点可能走到k号点

链状	c[k]=k号点的排位
树状	c[k]=k号点的深度
有向图	c[k]如何求解?

```cpp
const int N=1009;
bitset<N> G[N];
```

```cpp
    for(int u=1;u<=n;++u)G[u][u]=1;
    for(int u=1;u<=n;++u){
        int k,v;
        cin>>k;
        while(k--){
            cin>>v;
            G[u][v]=1;
        }
    }
    floyd();
```

```
void floyd(){
	for(int u=1;u<=n;++u)
		for(int v=1;v<=n;++v)
			if(            ) G[v]|=G[u];
	ans=0;
	for(int u=1;u<=n;++u){
		
		
		
		
	}
}
```

总结

请同学写总结：如何想到解法

简化问题

用极简形式2点3点问题能启发出递推式

单点对期望贡献

链状形态使用单点对期望贡献效果最佳

链状整体思维保留了部分全局观

2,3点整体性较弱，局部递推视野不宽阔

快快编程1806

潜在思路

A	简化问题	不能修改，静态求危险路径
B	简化问题	原始共1条危险路,2条,...
C	补集转换	补集转换，求不危险路径

算法步骤

危险边当做断边

求出各个连通块重新编号

补集转换

连通块块内路径都不危险

x号连通块大小为C_x时，块内路径数 $C_x*(C_x-1)/2$

每种修桥方案会连通两个连通块a和b
增加路径数C_a*C_b
求最优方案

点对距离和

一棵树有n个节点,n-1条边边长均为1。
求所有点对的距离总和是多少?

点对距离和

A	枚举点对	$O(n^2)$对枚举对象，再算距离

B	枚举起点	O(n)个枚举对象
	快速换根	u号为起点的单源多汇距离要1次DFS
		u为起点的距离和记作f[u]
		把起点从u号换到儿子v号时 f[v]对比f[u] 少了sz[v],多了n-sz[v]
	O(1)换根	f[v]=f[u]-sz[v]+n-sz[v]

点对距离和

C

个体贡献	每条边在距离总和里共计算几次

树上所有边都是桥	对于边e(u,v),进过e的路径两端必定1端在v这边,另1端在u这边

假定u是v的父节点
v这边共sz[v]点
u这边共n-sz[u]点

随机路径

树上共n点，随机取两个不同点u,v,
求dst(u,v)的期望

随机2点路径

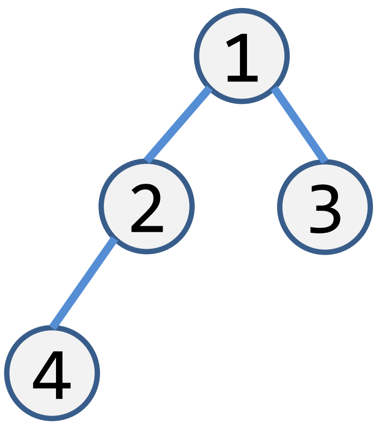
树上共n点，随机取两个不同点u,v，
求u到v距离d(u,v)的期望

期望定义	E[d(u,v)] = sum {选中(x,y)概率*d(x,y)} = 1/n * 1/(n-1) *2*点对距离和

随机遍历

按照1到n的随机排列顺序访问树上n个节点，
求每次随机访问的路径总长度的期望

按照1到n的全排列中的每种顺序访问树上n个节点，
求所有访问的路径总长度的总和



排列1234

总长=6

随机遍历

个体贡献	在每轮访问求路径总长里 "个体"选什么?点/边/路径?
	"个体"选路径
	有多少种排列里u后面紧接着出现v? (n-1)!
	特定的路径起点u到终点v 出现在某轮随机访问中的概率是1/n

每对点对的路径在访问中出现的概率2/n

E[单轮访问总长]=所有点对的距离和*2/n

随机3点路径

树上共n点，随机取3个不同点x,y,z
求距离d(x,y)+d(y,z)+d(z,x)的期望

快快编程1807

思路分析

一棵树n点，任意相隔1个点的2点连1条新边，
连接两个有公共邻居的点，问所有点对的距离之和。

重要思想：树加边不真的去加，还是用树结构计算

新边和旧边造成路径不唯一
注意力集中在旧边

新路径长度大概是旧路一半

原树中长度p为偶数的路径两端点在新图里距离p/2

原树中长度q为奇数的路径两端点在新图里距离(q+1)/2

思路分析

首先dfs求出深度d[],子树大小sz[]

原图点对距离和 x=sum_u sz[u]*(n-sz[u])

新图点对距离和 y
=原图偶数距离和/2+(原图奇数距离和+奇数距离路径数)/2

=(原图偶数距离和+原图奇数距离和+奇数距离路径数)/2
=(原图点对距离和x+奇数距离路径数)/2

其中奇数距离路径数怎么算?

其中奇数距离路径数=两端点深度一奇一偶的路径数
=奇数深度点数*偶数深度点数

快快编程作业

1804

1805

1806

拓展题	1807,1808,1809