目录

**[位运算](#_Toc21154_WPSOffice_Level1)** **[2](#_Toc21154_WPSOffice_Level1)**

**[1. 快速幂](#_Toc710_WPSOffice_Level1)** **[2](#_Toc710_WPSOffice_Level1)**

**[2. 64位整数乘法](#_Toc9861_WPSOffice_Level1)** **[3](#_Toc9861_WPSOffice_Level1)**

**[3. 最短Hamilton路径](#_Toc11007_WPSOffice_Level1)** **[3](#_Toc11007_WPSOffice_Level1)**

[1) 关心哪些点被用过](#_Toc710_WPSOffice_Level2) [3](#_Toc710_WPSOffice_Level2)

[2) 目前停在哪个点上](#_Toc9861_WPSOffice_Level2) [3](#_Toc9861_WPSOffice_Level2)

[第一维状态数 2^20](#_Toc11007_WPSOffice_Level2) [3](#_Toc11007_WPSOffice_Level2)

[第二维状态数 20](#_Toc18528_WPSOffice_Level2) [3](#_Toc18528_WPSOffice_Level2)

[2^20 \* 20 = 2 \* 10^7](#_Toc16897_WPSOffice_Level2) [3](#_Toc16897_WPSOffice_Level2)

[二、小知识：](#_Toc10564_WPSOffice_Level2) [4](#_Toc10564_WPSOffice_Level2)

**1.虚拟空间**

**2.cin优化**

**3.成对变化**

**[4. lowbit运算](#_Toc18528_WPSOffice_Level1)** **[5](#_Toc18528_WPSOffice_Level1)**

**位运算**

1. **快速幂**

int poww(int a,int b)

{

int ans=1,base=a;

while(b!=0)

{

if(b&1!=0) //b的最后一位不为0

{

ans\*=base;

}

base\*=base;

b>>=1;

}

return ans;

}

long long ksm(long long a,long long b) //快速幂a的b次{

long long ans=1;

while(b>0)

{

if(b%2==1)ans\*=a,ans%=mod;

a\*=a;

a%=mod;

b/=2;

}

return ans;

}

1. **64位整数乘法**

#include <bits/stdc++.h>

using namespace std;

typedef unsigned long long ull;

int main(){

ull a,b,p;

cin>>a>>b>>p;

ull res = 0;

while(b){

if(b&1) res = (res + a) % p;

a = a \* 2 % p;

b>>= 1;

}

cout<<res<<endl;

return 0;

}

1. **最短Hamilton路径**
2. 关心哪些点被用过
3. 目前停在哪个点上

第一维状态数 2^20

第二维状态数 20

2^20 \* 20 = 2 \* 10^7

**f[state][j]** state 表示哪些点被用过 j 表示现在在哪个点上

f[state][j] = f[state\_k][k] + weight[k][j];   
//state 表示哪些点被 遍历过的集合

**· State如何表示 ? 状态压缩 ！**

0 表示没走过(不在集合内) 1表示在集合内

例 ：走过0 1 4点 state = 10011

#include <iostream>  
#include <cstring>  
#include <algorithm>  
  
using namespace std;  
  
const int maxn = 1<<20;  
int f[maxn][25],val[25][25];  
int n;  
int main(){  
 ios::sync\_with\_stdio(false); //是否兼容stdio的开关  
 cin.tie(0); //tie是将scanf和cin两个stream绑定的的函数  
 cin>>n;  
 for(int i=0;i<n;i++)  
 for(int j=0;j<n;j++) cin>>val[i][j];  
  
 memset(f,0x3f,sizeof f);  
 f[1][0] = 0;  
 for(int i=0;i<1<<n;i++){  
 for(int j=0;j<n;j++){  
 if(i >> j & 1)  
 for(int k=0;k<n;k++){  
 if((i^(1<<j))>>k & 1)  
 f[i][j] = min(f[i][j],f[i^(1<<j)][k]+val[k][j]);  
 }  
 }  
 }  
 cout<<f[(1<<n)-1][n-1]<<endl;  
 return 0;  
}

二、小知识：

c++ 默认栈空间大小 4M = 2^22 约等于 4e6

栈空间 系统自动分配

堆空间 用户分配释放 静态变量 static 全局变量会放在堆里

所以习惯将变量放全局变量

1. **cin优化**

ios::sync\_with\_stdio(false); //是否兼容stdio的开关  
cin.tie(0); //tie是将scanf和cin两个stream绑定的的函数

1. 成对变换 用异或来实现配偶

0 1

2 3

4 5

...

n n+1

互相为配偶

n^1 = n+1 (n的配偶)

(n+1)^1 = n (n+1的配偶)

**n为偶数 nXOR1 = n+1 n为奇数 nXOR1 = n-1**

应用：图论邻接表边集的存储 最小费用流

1. lowbit运算

求出一个整数n在二进制表示下 最低的一位1是哪个

Lowbit(1110011000) = 1000

 110 -> 10

int lowbit(int n){  
 return n&(-n);  
}