

# 状压dp

---

## 人员

---

石宇赫、王崇宇、窦浩轩、李佳声、蔡云翔、王梓同、孙乐涵 到课

## 作业

---

<https://www.luogu.com.cn/contest/167200>

A、B 必做，C 选做

## 课堂表现

---

同学们整体代码实现能力都不是很好，还需要多练代码能力

## 课堂内容

---

判断第  $i$  位是否为1:  $(x \gg i) \& 1$

$x$  的第  $i$  位变为1:  $x \mid (1 \ll i)$

$x$  的第  $i$  位变为0:  $x \& (\sim(1 \ll i))$

把 $x$ 的最后一个1去掉:  $x \& (x - 1)$

### U111875 最短Hamilton路径

$f[i][j]$ : 所有走过点的状态为  $i$ ，以  $j$  结尾时，路径长度的最小值

```

#include <bits/stdc++.h>
using namespace std;
const int N = 20;
int f[1<<N][N];
int w[N][N];
int main() {
    int n; cin >> n;
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) cin >> w[i][j];
    }

    memset(f, 0x3f, sizeof(f));
    f[1][0] = 0;

    for (int i = 2; i < (1<<n); ++i) {
        for (int j = 0; j < n; ++j) {
            if ((i>>j) & 1) { // 求 f[i][j]: 由 f[i-(1<<j)][k] 转移而来
                for (int k = 0; k < n; ++k) {
                    if (k == j) continue;
                    if ((i>>k) & 1) {
                        f[i][j] = min(f[i][j], f[i-(1<<j)][k] + w[k][j]);
                    }
                }
            }
        }
    }
    cout << f[(1<<n)-1][n-1] << endl;
    return 0;
}

```

## U204941 蒙德里安的梦想

$f[i][j]$ : 第  $i$  列的二进制状态为  $j$

1. 输入  $n, m$

2. 初始化  $vec\ 0 \sim (1<<n)-1$

$bool\ check(int\ x, int\ n)\ \{ \quad // \text{判断}x\text{的}n\text{位二进制是否没有连续奇数个}0$

```

}
for (int i = 0; i < (1<<n); ++i) {
    for (int j = 0; j < (1<<n); ++j) {
        if ((i&j)==0 && check(i|j, n)) {
            vec[i].push_back(j);
        }
    }
}
}

```

3. 求  $f[i][j]$

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int N = 11;
bool st[1<<N];
vector<int> vec[1<<N];
LL f[N+5][1<<N];

bool check(int x, int n) {
    int cnt = 0;
    for (int i = 0; i < n; ++i) {
        if ((x>>i)&1) {
            if (cnt&1) return false;
            cnt = 0;
        } else ++cnt;
    }
    if (cnt&1) return false;
    return true;
}

void init(int n) {
    for (int i = 0; i < (1<<n); ++i) {
        st[i] = false;
        if (check(i, n)) st[i] = true;
    }

    for (int i = 0; i < (1<<n); ++i) {
        vec[i].clear();
        for (int j = 0; j < (1<<n); ++j) {
            if (i&j) continue;
            if (st[i|j]) vec[i].push_back(j);
        }
    }
}

int main()
{
    int n, m;
    while (true) {
        cin >> n >> m;
        if (!n && !m) break;
        init(n);
        memset(f, 0, sizeof(f));
        f[0][0] = 1;
        for (int i = 1; i <= m; ++i) {
            for (int j = 0; j < (1<<n); ++j) {
                for (int k : vec[j]) {
                    f[i][j] += f[i-1][k];
                }
            }
        }
    }
}

```

```
    }  
  }  
}  
cout << f[m][0] << endl;  
}  
return 0;  
}
```