```cpp
#include "../bits/stdc++.h"
// 最大独立集合問題 V=40 ぐらいまで
// 最小頂点被覆問題と等価
// 補グラフの最大クリーク問題と等価
// 2部グラフ上の最大独立集合問題 = 頂点数 - 最大マッチング
// verified: https://codeforces.com/contest/1105/submission/50254980
class MaximumIndependentSet
{
    int n, ret = 0;
    std::vector<long long> g;

    int dfs(long long remain, int cnt)
    {
        for (bool update = true; update;)
        {
            update = false;
            for (int i = 0; i < n; i++)
                if (remain & (1LL << i))
                {
                    int deg = __builtin_popcountll(remain & g[i]);
                    if (deg <= 1)
                    {
                        cnt++;
                        remain &= ~((1LL << i) | g[i]);
                        update = true;
                    }
                }
        }
        ret = std::max(ret, cnt);
        if (remain)
        {
            int k = __builtin_ctzll(remain);
            dfs(remain & ~(1LL << k), cnt);
            dfs(remain & ~(g[k] | (1LL << k)), cnt + 1);
        }
    }

  public:
    MaximumIndependentSet(int _n) : n(_n), g(n, 0) {}
    void addEdge(int u, int v)
    {
        g[u] |= (1LL << v);
        g[v] |= (1LL << u);
    }
    int solve()
    {
        dfs((1LL << n) - 1, 0);
        return ret;
    }
};
```