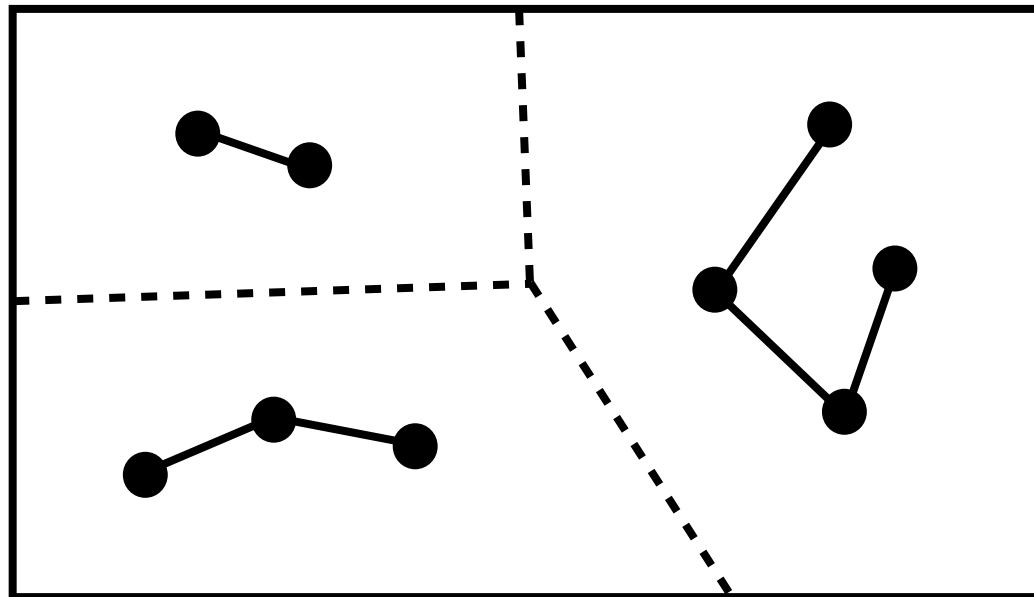


ALGORITMOS AVANZADOS SOBRE GRAFOS

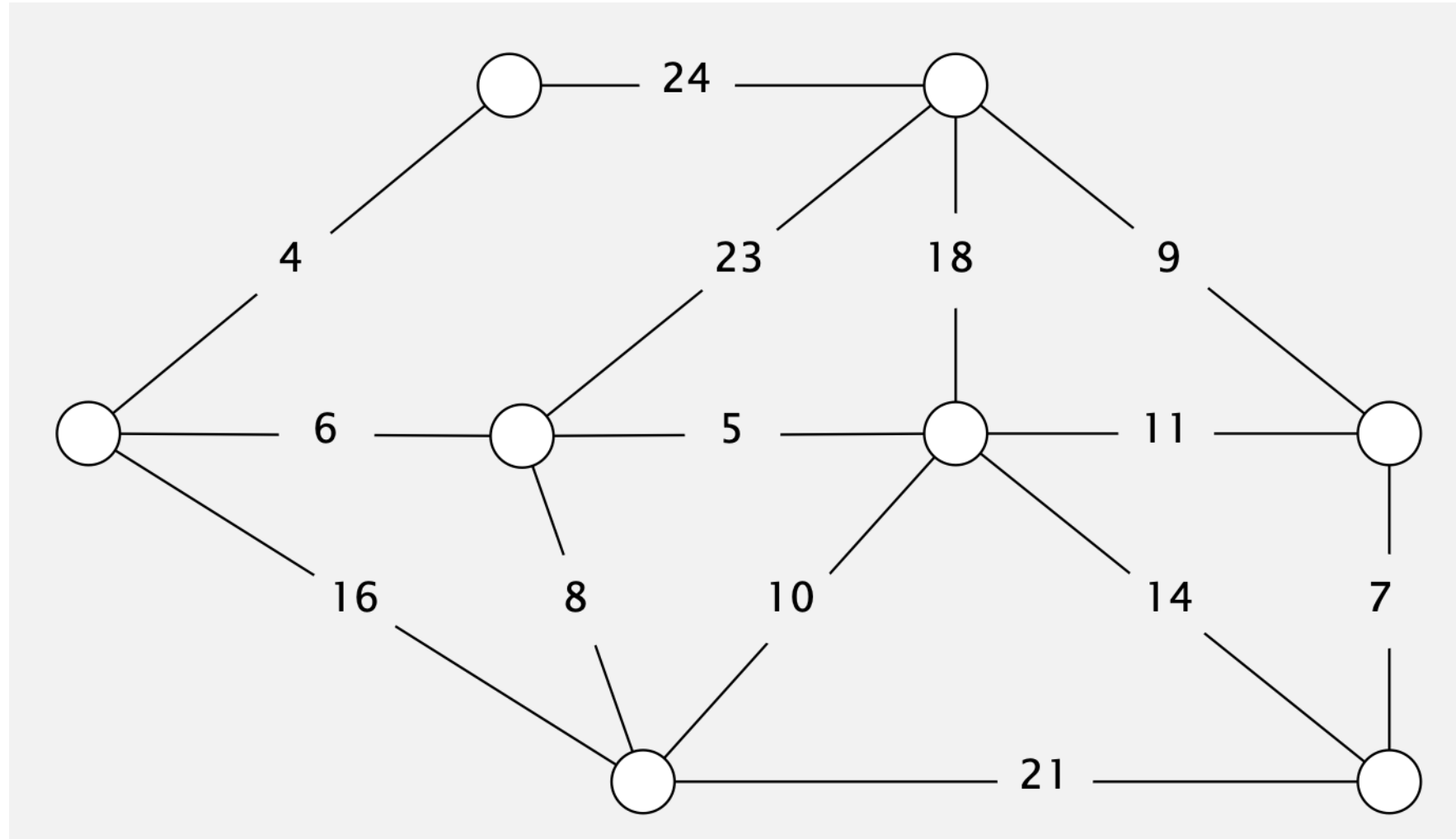
ARMs y Matching en grafos bipartitos

Union-find

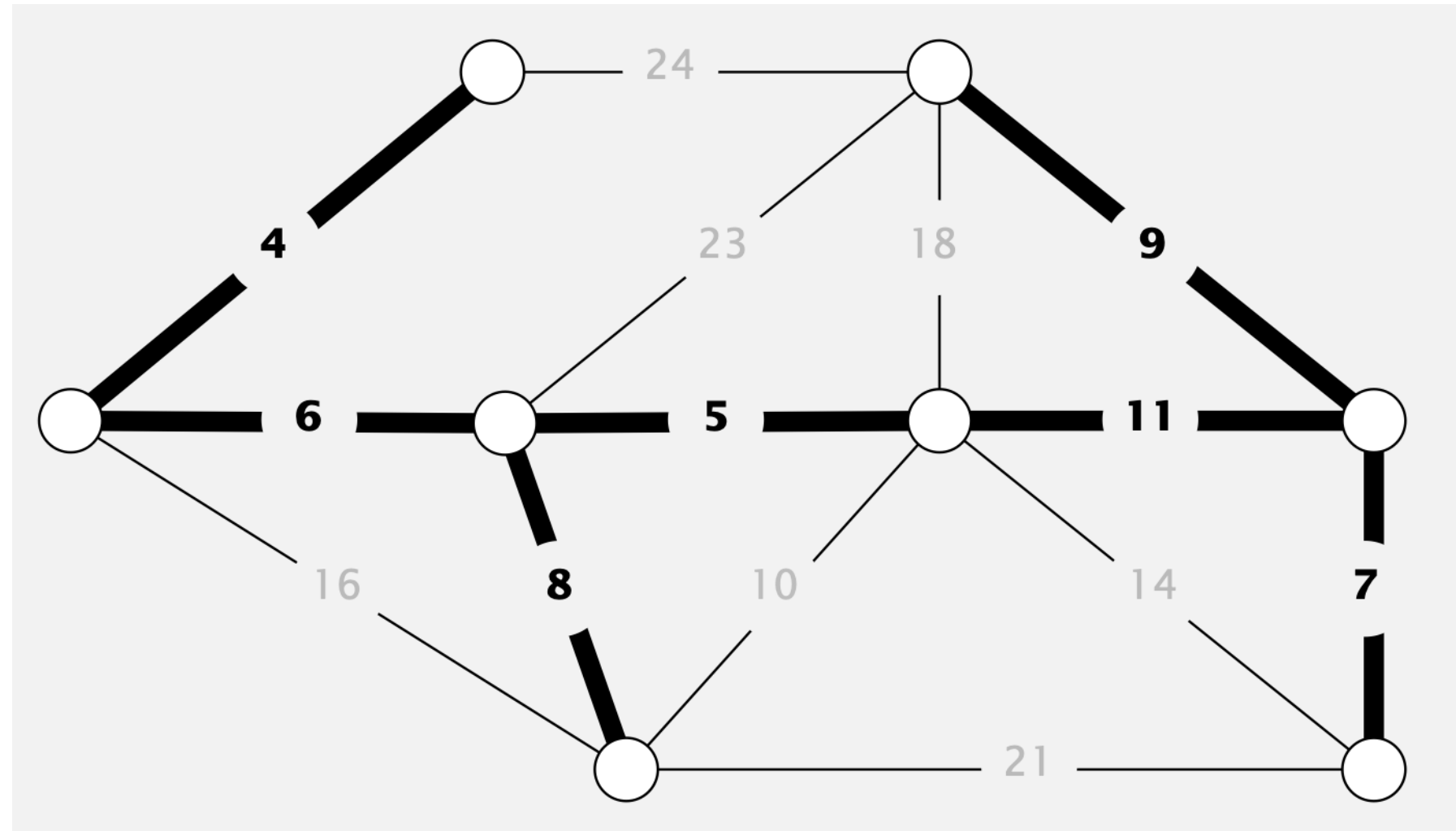


```
struct UFDS {  
    vector<int> p;  
    int numSets;  
    UFDS(int n) : p(n, 0), numSets(n) {  
        for (int i = 0; i < n; ++i) p[i] = i;  
    }  
    int find(int x) {  
        return (p[x] == x) ? x : p[x] = find(p[x]);  
    }  
    void merge(int x, int y) {  
        int i = find(x), j = find(y);  
        if (i == j) return;  
        p[i] = j;  
        --numSets;  
    }  
};
```

Minimum Spanning Tree



Minimum Spanning Tree



Algoritmo de Kruskal

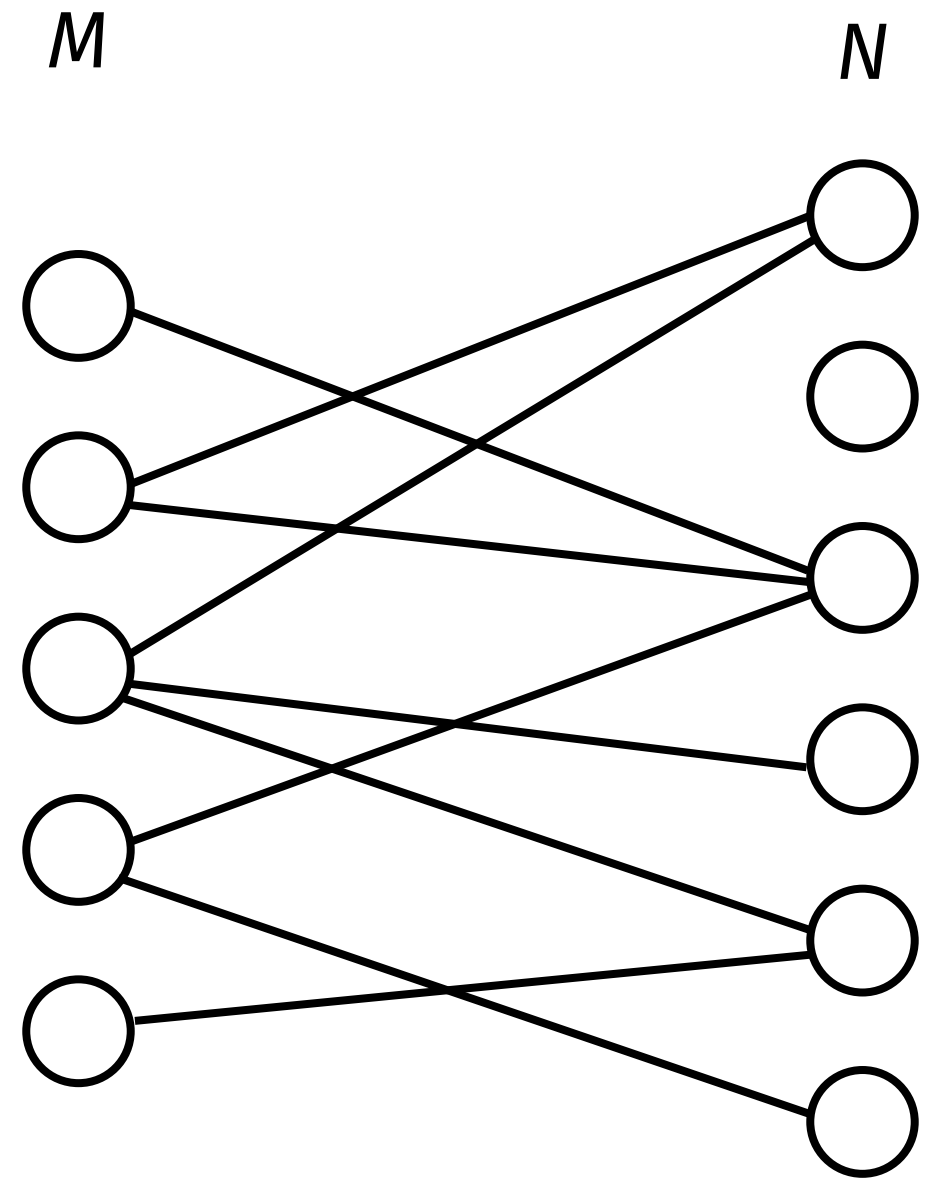
```
using arista = pair<int, pair<int, int>>; // < coste, extremos >

vector<arista> aristas;
// generar aristas y guardarlas en el vector

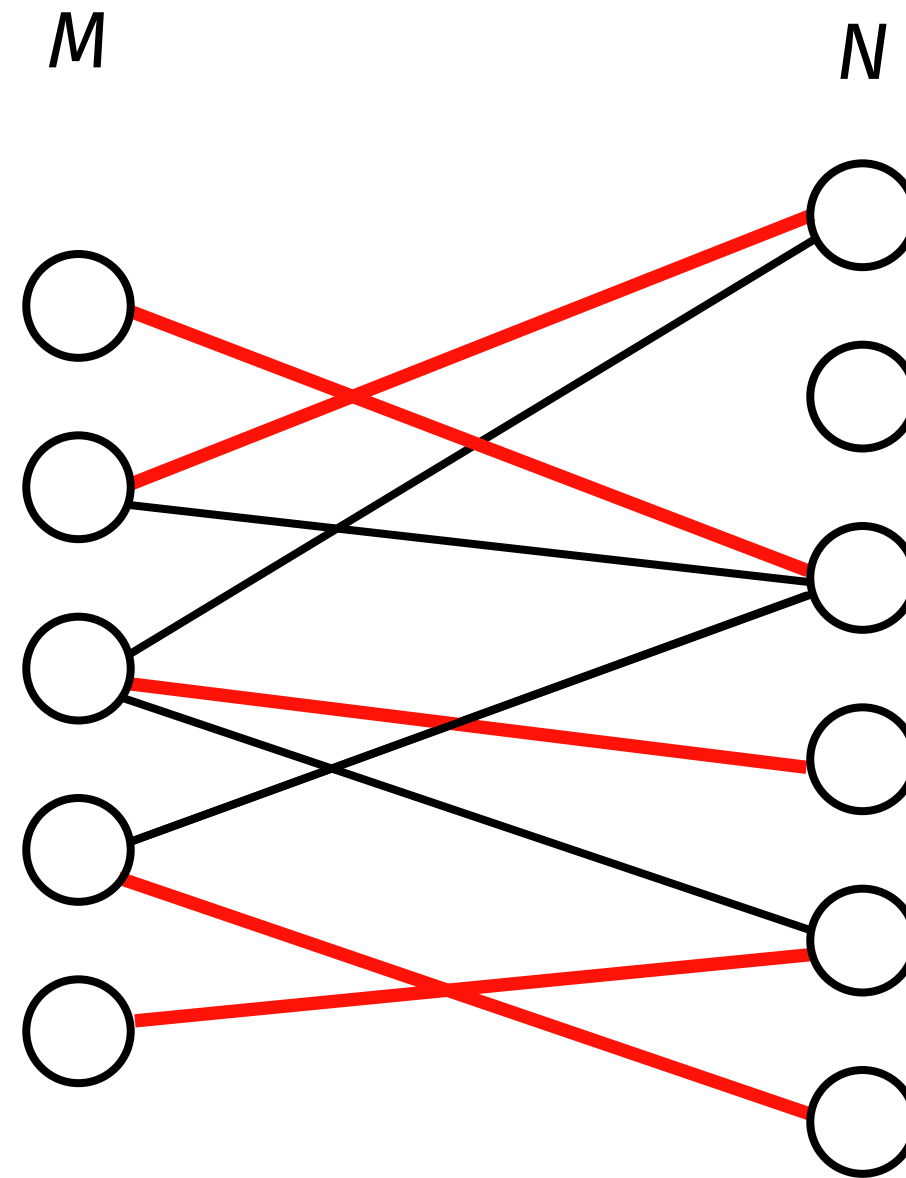
// ordenamos las aristas de menor a mayor
sort(aristas.begin(), aristas.end());

// Kruskal
UFDS uf(V);
int coste = 0;
for (auto ar : aristas) {
    if (uf.find(ar.second.first) != uf.find(ar.second.second)) {
        uf.merge(ar.second.first, ar.second.second);
        coste += ar.first;
        if (uf.numSets == 1) break;
    }
}
cout << coste << '\n';
```

Max Cardinality Bipartite Matching



Max Cardinality Bipartite Matching

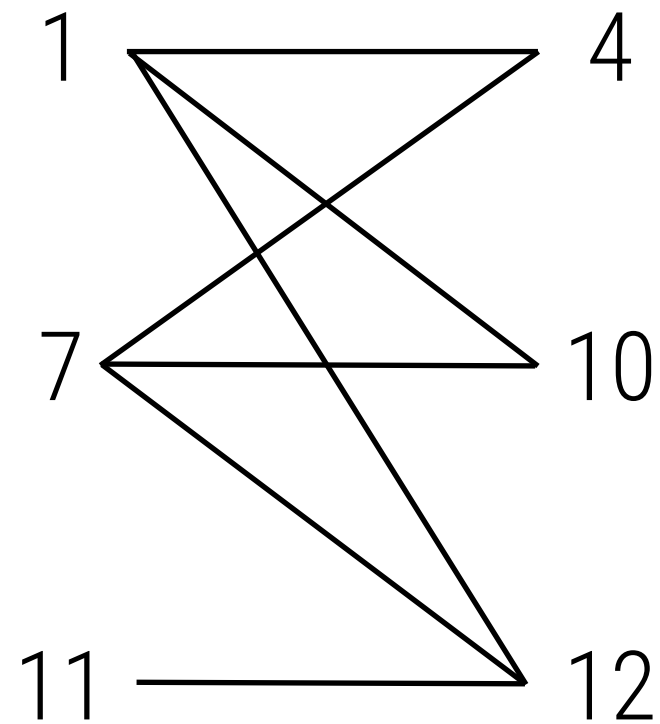


Max Cardinality Bipartite Matching

- Dado un conjunto de números > 0 , formar el mayor número de parejas con números distintos tales que la suma de sus números sea un número primo.
- Por ejemplo, $\{ 1, 4, 7, 10, 11, 12 \}$

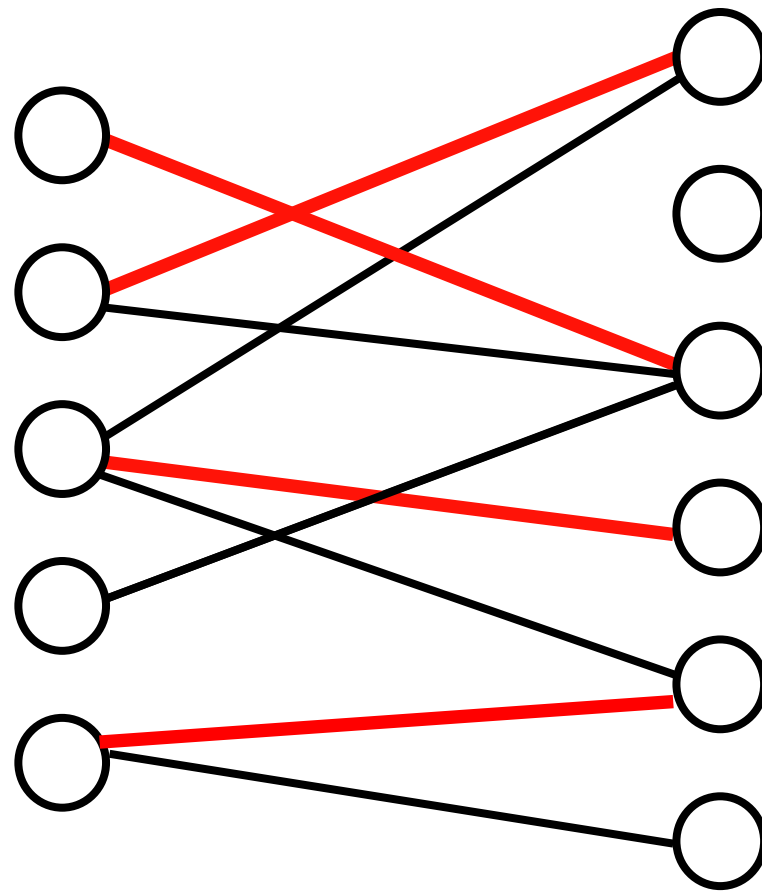
Max Cardinality Bipartite Matching

- Dado un conjunto de números > 0 , formar el mayor número de parejas con números distintos tales que la suma de sus números sea un número primo.
- Por ejemplo, $\{ 1, 4, 7, 10, 11, 12 \}$



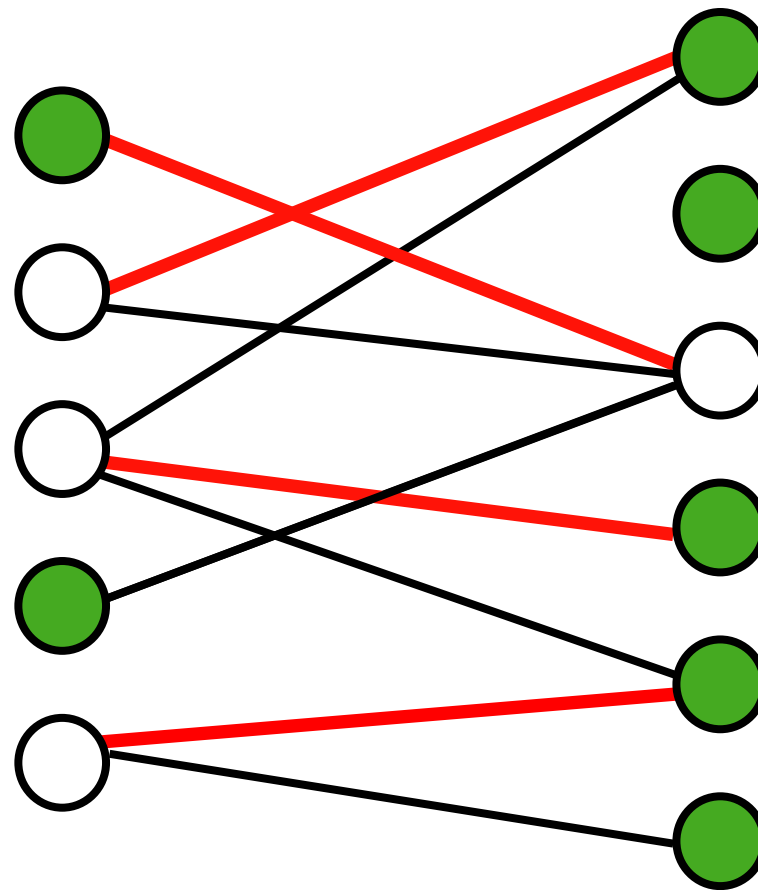
Max Independent Set

- Mayor conjunto de vértices tal que ningún par de vértices del conjunto representa una arista.
- $\text{MCBM} + \text{MIS} = V$ $\text{MIS} = V - \text{MCBM}$



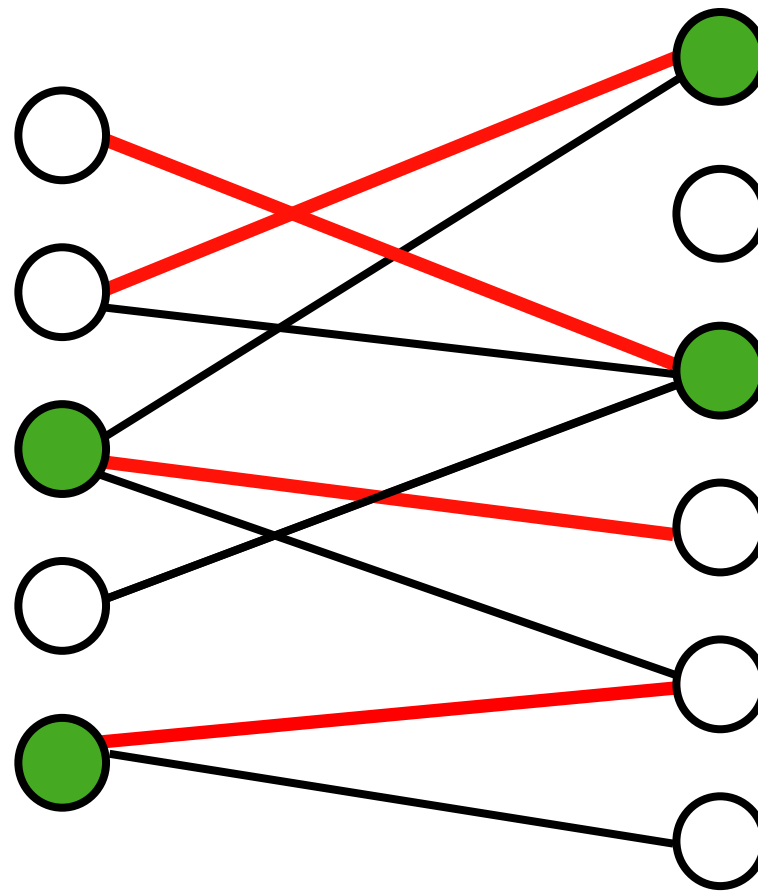
Max Independent Set

- Mayor conjunto de vértices tal que ningún par de vértices del conjunto representa una arista.
- $\text{MCBM} + \text{MIS} = V$ $\text{MIS} = V - \text{MCBM}$



Min Vertex Cover

- Menor conjunto de vértices tal que cada arista del grafo tiene al menos un extremo en el conjunto.
- MVC = MCBM



Augmenting Path Algorithm for MCBM

```
int M, N; // M parte izquierda, N parte derecha
vector<vector<int>> grafo; // dirigido, tamaño M
vector<int> match, vis;

int aug(int l) { // devuelve 1 si encuentra un augmenting path
                // para el matching M representado en match
    if (vis[l]) return 0;
    vis[l] = 1;
    for (auto r : grafo[l])
        if (match[r] == -1 || aug(match[r])) {
            match[r] = l;
            return 1;
        }
    return 0;
}

int berge_mcbm() {
    int mcbm = 0;
    match.assign(N + M, -1);
    for (int l = 0; l < M; l++) {
        vis.assign(M, 0);
        mcbm += aug(l);
    }
    return mcbm;
}
```

$O(VE)$

Hopcroft Karp: $O(\sqrt{V}E)$