```cpp
/*
    | Hopcroft-Karp algorithm |
    Desc: Maximum Bipartite in O(E*sqrt(V))
    Source: KawakiMeido
    State: Untested lmao
*/

int pairX[N],pairY[N],dist[N];
bool visX[N],visY[N];

bool BFS(){
    memset(visX,0,sizeof(visX));
    memset(visY,0,sizeof(visY));
    queue<int> q;
    for (int i=1; i<=n; i++){
        if (pairX[i] == 0){
            dist[i] = 0;
            q.push(i);
        }
        else dist[i] = INF;
    }
    dist[0] = INF;
    while (!q.empty()){
        int x = q.front();
        q.pop();

        visX[x] = true;
        for (auto y:adj[x]){
            int v = pairY[y];
            visY[y] = true;
            if (dist[v]==INF){
                dist[v] = dist[x]+1;
                q.push(v);
            }
        }
    }
    return (dist[0]!=INF);
}

bool DFS(int u){
    if (u == 0) return true;
    for (auto y:adj[u]){
        int v = pairY[y];
        if (dist[v] == dist[u]+1 && DFS(v)){
            pairX[u] = y;
            pairY[y] = u;
            return true;
        }
    }
    dist[u] = INF;
    return 0;
}

int Hopcroft_Karp(){
    int matching = 0;
    memset(pairX,0,sizeof(pairX));
    memset(pairY,0,sizeof(pairY));
    while (BFS()){
//          cout << dist[0] << endl;
        for (int i=1; i<=n; i++){
            if (pairX[i] == 0 && DFS(i)){
                ++matching;
            }
        }
    }
    return matching;
}
```