

Max Matching in Bipartite graph

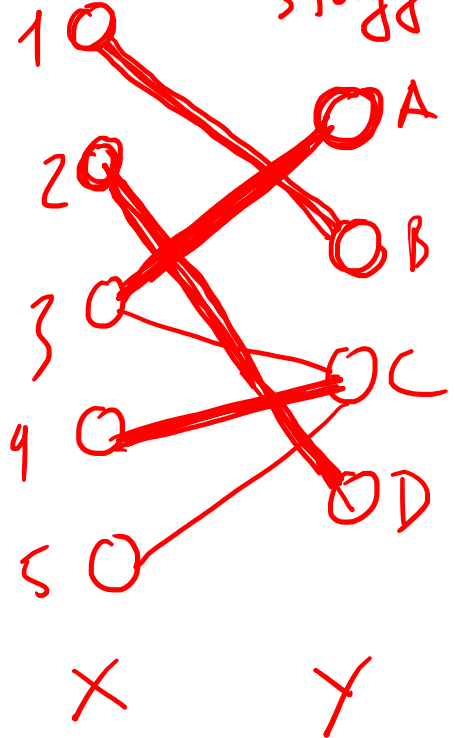
(Bài toán cặp ghép cực đại trên đồ thị 2 phía)

cho đồ thị 2 phía $G=(X,Y,E)$

Task

stagg

cặp ghép: tập con của E



Sao cho \forall có 2 cạnh nào chung đầu nút

(1, B)

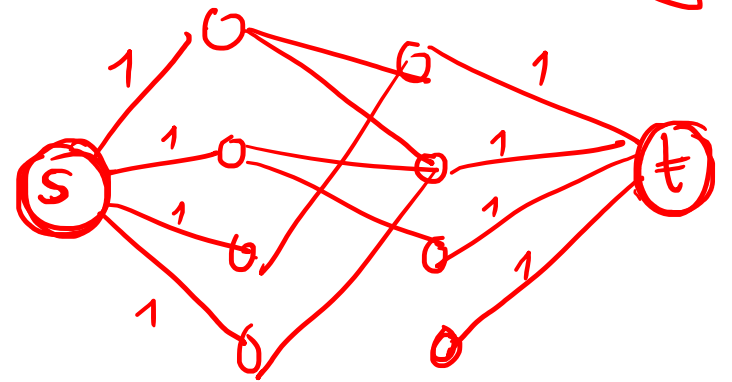
(3, A), (~~3, C~~)

(4, C)

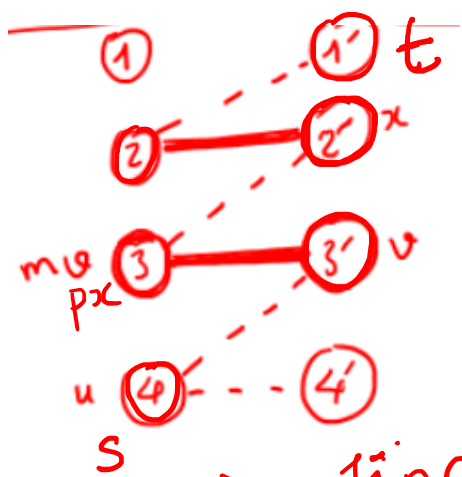
(2, D)

Mục tiêu: chọn tập
còn có số cạnh nhiều nhất

Thuật toán Hungary



- Xuất phát từ cặp ghép rỗng
- Loop
 - Tìm augmenting path P
 - Đảo trạng thái (chưa ghép / đã ghép) dọc theo P để tăng số cạnh tăng phiên bản
- Dừng nên không tìm được augmenting path



find augmenting path từ u

```

BFS( $u$ ) { //  $u$ : bên trái
     $Q = \text{empty queue}$ ; // lưu đỉnh bên phải
    for  $x \in T[u]$  do {
         $Q.\text{push}(x)$ ;  $p[x] = u$ 
    }
    while  $Q$  not empty do {
         $v = Q.\text{pop}()$ ;  $mu = \text{match}[v]$ ;
        if  $mu = -1$  then return  $v$ ;
        for  $x \in T[mu]$  do
            if  $x \neq r\text{Match}[mu]$  then {
                 $Q.\text{push}(x)$ ;  $p[x] = mu$ 
            }
    }
}
return -1; // not found.
}

```

Increase Matching(s, t) { // use $p[x]$

$x = t$;

while $p[x] \neq s$ do {

$px = p[x]$; $nx = r\text{Match}[px]$;

 // x và px ghép với nhau.

$\text{match}[x] = px$; $r\text{Match}[px] = x$;

$x = nx$;

} // ghép nốt x với $p[x]$.

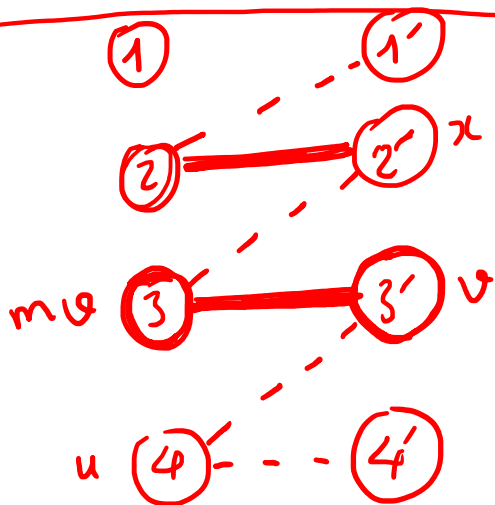
$\text{match}[x] = p[x]$; $r\text{Match}[p[x]] = x$;

}

D

Data structures.

- $match[i]$: là đỉnh bên trái được ghép với đỉnh i (i : bên phải)
- $rMatch[j]$: đỉnh bên phải được ghép với đỉnh j (j : bên trái)
- $T[j]$: dis các đỉnh bên phải kề với j (j : bên trái)
- $p[i]$: đỉnh cha của i trong quá trình BFS (i : bên phải)



$p[3'] = 4, match[3'] = 3$
 $p[2'] = 3, match[2'] = 2$
 $rMatch[3] = 3', rMatch[2] = 2'$
 $p[1'] = 2$
 $T[4] = \{3', 4'\}$

```

BFS(u) { // u: bên trái
    Q = empty queue; // lưu đỉnh bên phải
    for x ∈ T[u] do {
        Q.push(x); p[x] = u
    }
    while Q not empty do {
        v = Q.pop(); mv = match[v];
        if mv = -1 then return v;
        for x ∈ T[mv] do
            if x ≠ rMatch[mv] then {
                Q.push(x); p[x] = mv;
            }
    }
    return -1; // not found.
}

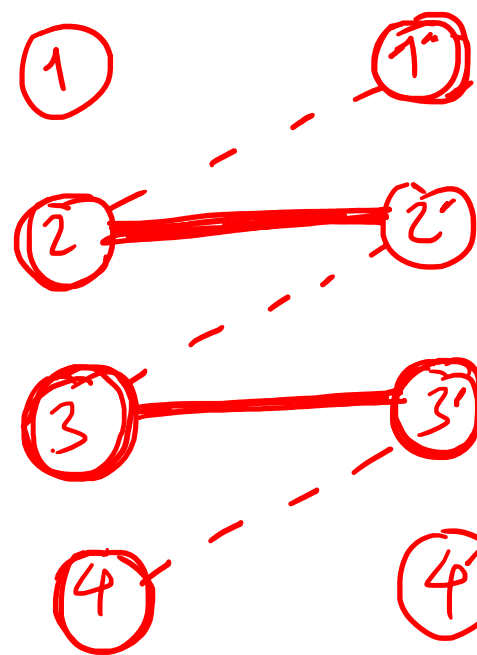
```

augmenting path:

- Xuất phát từ 1 đỉnh bên trái, đi qua 1 đỉnh bên phải theo cạnh chưa ghép, đi tiếp sang 1 đỉnh bên trái theo cạnh đã ghép, đi tiếp sang 1 đỉnh bên phải theo 1 cạnh chưa ghép, đi tiếp sang 1 đỉnh bên trái theo 1 cạnh đã ghép

- Dừng lại tại 1 đỉnh bên phải chưa được ghép

← chưa ghép



Cặp ghép hiện tại
(3, 3'), (2, 2')

↓ augmenting path

