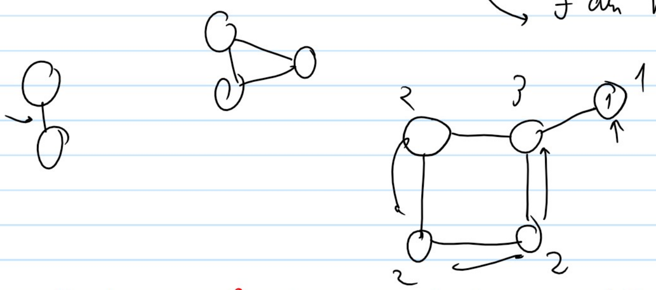
Buổi 3. Euler, Hamilton & Giải thuật tìm đường

1, Chu trình và đg đi Euler

- ↳ đơn đi qua ∀ cạnh (.) đồ thị → đồ thị G<V, E> có CT Euler → đồ thị Euler
- đg đi " , c cần quay lại xphát → đồ thị nửa Euler

Vô hướng: → Liên thông → ∀ đỉnh đều có deg = chẵn → Euler
→ ∃ đỉnh bậc lẻ → ≤ 2 đỉnh → nửa Euler



Nếu đồ thị nửa Euler → Xphát từ đỉnh bậc lẻ (có TT từ điểm min)

Có hướng → liên thông yếu → ∀ đỉnh đều có $deg^+ = deg^-$ → Euler
→ ∃ " $deg^+ \neq deg^-$ → Số lg đỉnh có lệch quá 2 e ↓ ≤ 2 → nửa Euler

Nửa Euler — Xphát từ đỉnh lẻ b bra lá nhỉ

Thuật toán tìm CT Euler

Bước 3 (Xây dựng chu trình/đường đi Euler bắt đầu từ đỉnh u):

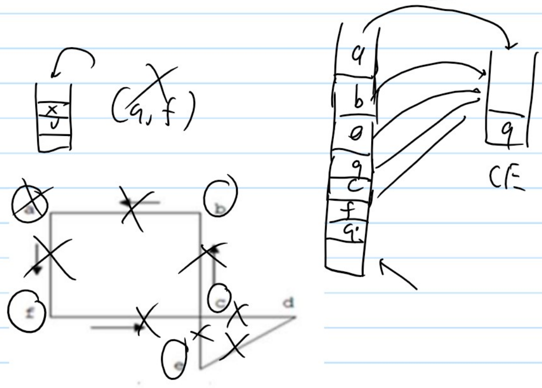
(3.1) Tạo mảng CE để ghi chu trình/ đường đi Euler và Stack để xếp các đỉnh sẽ xét. Xếp đỉnh u vào Stack;

(3.2) Xét đỉnh v nằm trên cùng của Stack và thực hiện:

- Nếu v là đỉnh cô lập thì lấy v ra khỏi Stack và đưa vào CE;
- Nếu v có đỉnh kề là x thì đưa x vào Stack sau đó xóa cạnh nối v với x;

(3.3) Quay lại (3.2) cho tới khi stack rỗng;

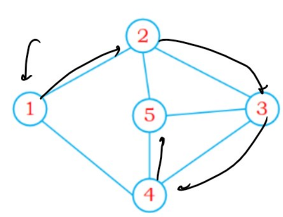
Bước 4: Xuất chu trình/đường đi Euler chứa trong CE theo thứ tự ngược lại.

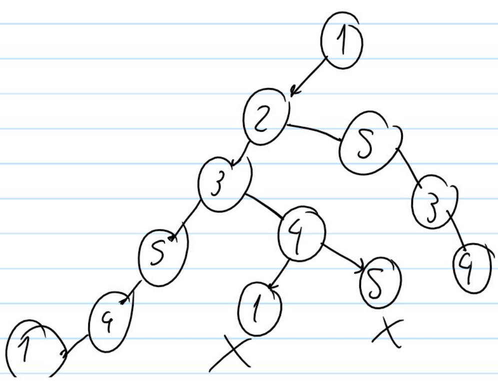


a b c d e f a

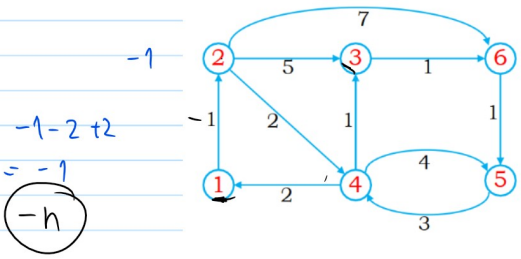
2, Hamilton

- CT đơn đi qua tất cả các đỉnh (.) đồ thị.
- đg đi → bđầu có thể ≠ kthúc





Dijkstra - Bellman Ford - Floyd

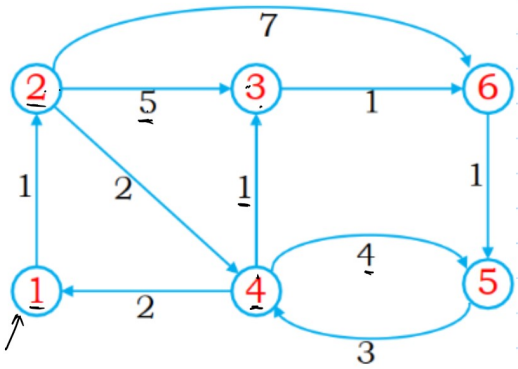


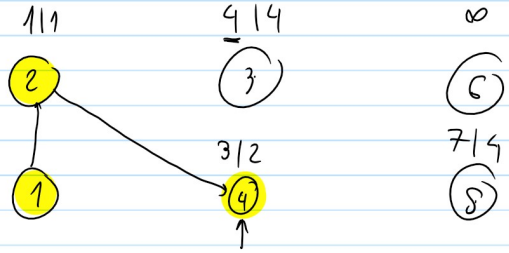
Dijkstra → KN: Tìm độ dài ngắn nhất từ 1 đỉnh → các đỉnh ≠ 0 đồ thị
→ YC: đồ thị ko có cạnh âm

Bellman - Ford → KN: Tìm độ dài ngắn nhất từ 1 đỉnh → các đỉnh ≠ 0 đồ thị
→ YC: đồ thị có thể có cạnh âm, ko có chu trình âm

Floyd → KN: tìm độ dài ngắn nhất giữa 2 đỉnh u, v bất kì đồ thị
→ YC: đồ thị có thể có cạnh âm, ko có chu trình âm

Dijkstra: CT thuật tham lam + gán nhãn tạm thời:





Bước lặp	Đỉnh 1	Đỉnh 2	Đỉnh 3	Đỉnh 4	Đỉnh 5	Đỉnh 6
Khởi tạo	0, 1	1, 1*	∞, 1	∞, 1	∞, 1	∞, 1
1	-	-	6, 2	3, 2*	∞, 1	8, 2
2	-	-	4, 4*	-	7, 4	8, 2
3	-	-	-	-	7, 4	5, 3*
4	-	-	-	-	6, 6*	-
5						

5 ← 6 ← 3 ← 4 ← 2
↑
1

truoc[1] = 0

truoc 2 = 1
" 3 = 4
" 4 = 2
" 5 = 6
" 6 = 3

1 → 5
end = 5
while (end != 1) {
    Cout << end;
    end = truoc[end];
}

Bellman - Ford

Lặp : Ktra cặp đỉnh i ; j   n-2 lần thì Ktra

$d[j] = 5$

$d[i] + a[i][j] = 4$ → update cho đỉnh j

Bellman-Ford(s){

Bước 1 (Khởi tạo):

for $(v \in V)${ //Sử dụng s gán nhãn cho các đỉnh còn lại

$d[v] = a(s, v)$;

$truoc[v] = s$;

}

Bước 2 (Lặp):   n-2 lần

$d[s] = 0$; k=1

while $(k \le n-2)${

for $(v \in V \backslash \{s\})${

for $(u \in V)${

if $(d[v] > d[u] + a(u, v))${

$d[v] = d[u] + a(u, v)$;

$truoc[v] = u$;

}

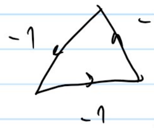
}

}

}

$d[v] >$

$d[2] > d[3] + a[3][2]$?



-3

-6

-9

relaxation

k → n-2

n   n-1

Floyd : quy hoạch động : $d[i][j]$

$d[i][j] > d[i][k] + d[k][j]$?

update : $d[i][j] = d[i][k] + d[k][j]$;

$tiep[i][j] = k$

```
for (k ∈ V) {
  for (i ∈ V) {
    for (j ∈ V) {
        d[i][j] > d[i][k] + d[k][j] ?
        d[i][j] = d[i][k] + d[k][j];
        tiep[i][j] = k
    }
  }
}
```