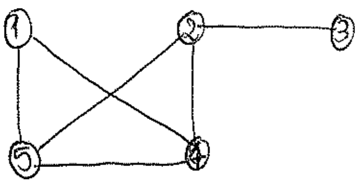
* DFS:

Nguyễn Trọng Minh Hồng Phước



B_1: Khởi tạo:

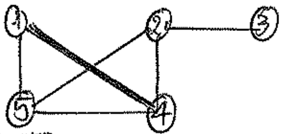
	1	2	3	4	5
CX	0	0	0	0	0
LV	-1	-1	-1	-1	-1

B_2: Xuất phát từ 1 ⇒ CX [1] = 1

	1	2	3	4	5
CX	1	0	0	0	0
LV	-1	-1	-1	-1	-1

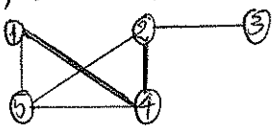
B_3: Từ 1 có đường đi đến 4,5 ⇒ xét đỉnh 4
⇒ CX [4] = 1, LV [4] = 1

	1	2	3	4	5
CX	1	0	0	1	0
LV	-1	-1	-1	1	-1



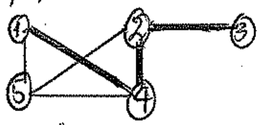
B_4: Từ 4 có đường đi đến 1,2,5. Mà CX [1] = 1 nên bỏ qua 1. Xét đỉnh 2 ⇒ CX [2] = 1, LV [2] = 4

	1	2	3	4	5
CX	1	1	0	1	0
LV	-1	4	0	1	0



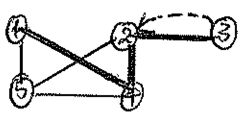
B_5: Từ 2 có đường đi đến 5,4,3. Mà CX [4] = 1 nên bỏ qua 4. Xét đỉnh 3 ⇒ CX [3] = 1, LV [3] = 2

	1	2	3	4	5
CX	1	1	1	1	0
LV	-1	4	2	1	-1



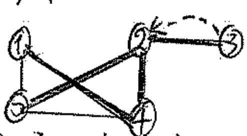
B_6: Từ 3 có đường đi đến 2, mà CX [2] = 1 nên ta quay lui về 2.

	1	2	3	4	5
CX	1	1	1	1	0
LV	-1	4	2	1	-1



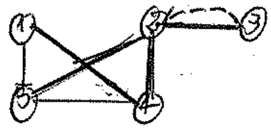
B_7: Từ 2 có đường đi đến 4,5,3. Mà CX [3] = CX [4] = 1 nên bỏ qua 3,4. Xét đỉnh 5 ⇒ CX[5] = 1, LV[5] = 2

	1	2	3	4	5
CX	1	1	1	1	1
LV	-1	4	2	1	2



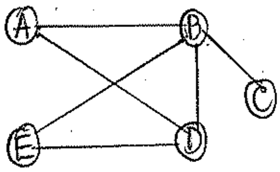
B_8: Lúc này tất cả các đỉnh đã xét ⇒ kết thúc thuật toán.

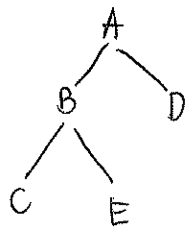
Vậy duyệt DFS từ 1:



	1	2	3	4	5
CX	1	1	1	1	1
LV	-1	4	2	1	2

* BFS:





Qui ước: {A, B, C, D, E} ⇔ {0, 1, 2, 3, 4}.

B_1: Khởi tạo:

	0	1	2	3	4
CX	0	0	0	0	0
LV	-1	-1	-1	-1	-1

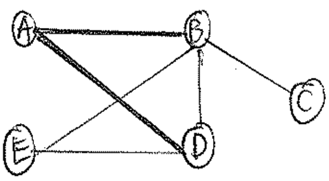
	0	1	2	3	4
Queue					

B_2: Xuất phát từ A ⇒ CX[0] = 1, đẩy 0 vào Queue

	0	1	2	3	4
CX	1	0	0	0	0
LV	-1	-1	-1	-1	-1

	0	1	2	3	4	5
Q	0					

B_3: Queue chưa rỗng, lấy từ Queue ra 0 (đỉnh A). A kề với B, D ⇒ đưa 1, 3 vào queue.



	0	1	2	3	4
CX	1	~~0~~ 1	0	~~0~~ 1	0
LV	-1	0	-1	0	-1

	0	1	2	3	4
Q	~~0~~	1	3		

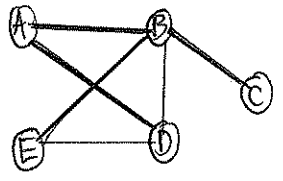
⇒ CX [1] = CX[3] = 1
LV [1] = LV[3] = 0

B_4: Queue chưa rỗng, lấy từ Queue ra 1 (đỉnh B) ⇒ CX[1] = 1, LV[1] = 0

Đỉnh B kề với A, C, E. Mà CX[0] = 1 nên bỏ qua A. Xét đ² C, E ⇒ Đưa 2, 4 vào Queue.

⇒ CX[2] = CX[4] = 1

LV[2] = LV[4] = 1



	0	1	2	3	4
CX	1	1	1	1	1
LV	-1	0	1	0	1

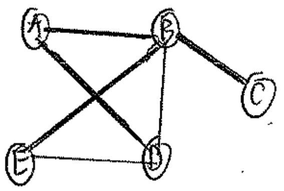
	0	1	2	3	4
Q	~~0~~	~~1~~	3	2	4

B_5: Queue chưa rỗng, lấy từ Queue ra 3 (đỉnh D) ~~⇒ CX[3] = 1, LV[3] = 0~~

Đỉnh D kề với A, B, E. Mà CX[0] = CX[1] = CX[4] = 1 nên bỏ qua đ² D.

B_6: Queue chưa rỗng, lấy từ Queue ra 2 (đỉnh C). Đỉnh C kề với B mà CX[1] = 0 nên bỏ qua đ² C.

B_7: Queue chưa rỗng, lấy từ Queue ra 4 (đỉnh E). Đỉnh E kề với B, D mà CX[1] = CX[3] = 1 nên bỏ qua E.

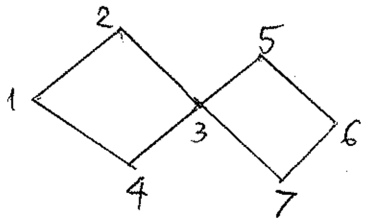


	0	1	2	3	4
CX	1	1	1	1	1
LV	-1	0	1	0	1

	0	1	2	3	4
Q	~~0~~	~~1~~	~~3~~	~~2~~	~~4~~

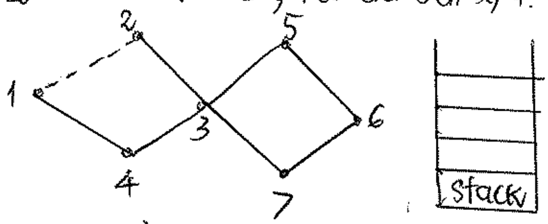
⊛ EULER:

"Khi nào k⁰ còn đường thì đưa đỉnh đó vào Stack".

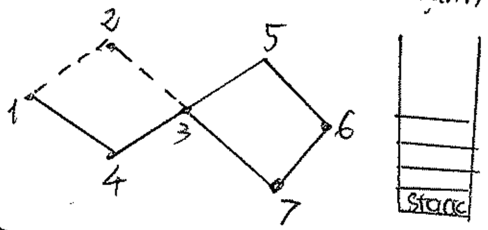


B_1: Khởi tạo Stack = ∅.

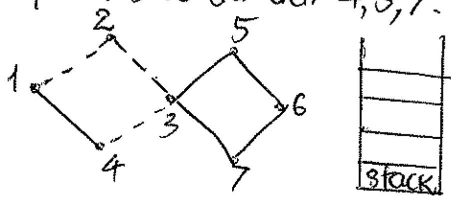
B_2: Bắt đầu từ 1, 1 có đđ đến 2, 4. Xét (1,2) và xoá (1,2).



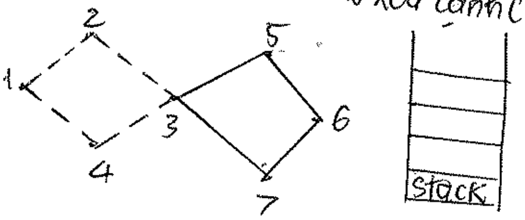
B_3: Từ 2 có đđ đến 3. Xét & xoá cạnh (2,3).



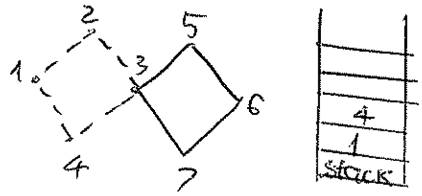
B_4: Từ 3 có đđ đến 4, 5, 7. Xét & xoá (3,4).



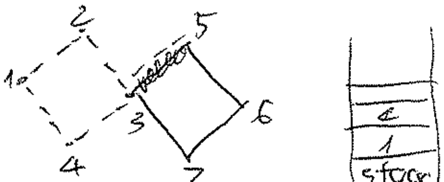
B_5: Từ 4 có đđ đến 1. Xét & xoá cạnh (1,4).



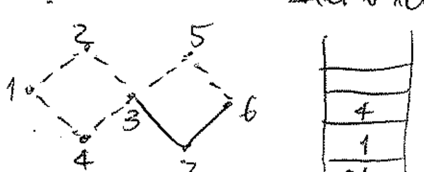
B_6: Tại 1 k⁰ còn đđ ⇒ đưa 1 vào stack và quay về 4. Tương tự đưa 4 vào stack và quay về 3.



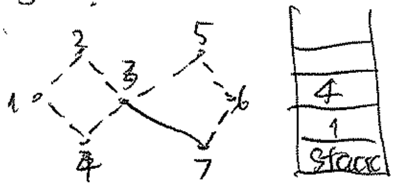
B_7: Tại 3 có đđ đến 5, 7. Xét và xoá (3,5).



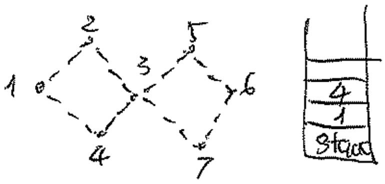
B_8: Tại 5 có đđ đến 6. Xét & xoá (5,6)



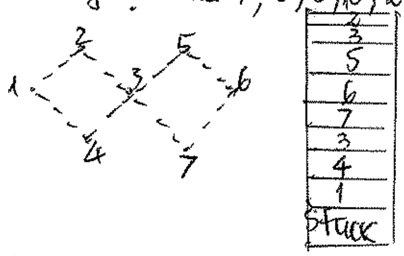
B_9: Tại 6 có đt đến 7. Xét & xoá (6,7)



B_{10}: Tại 7 có đt đến 3. Xét & xoá (7,3).

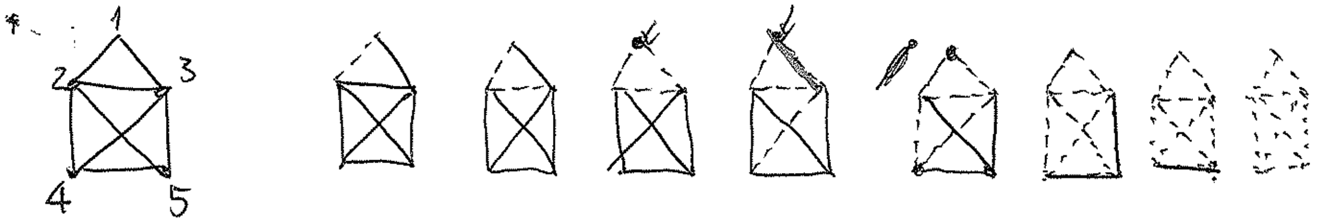


B_{11}: Tại 3 k° còn đt ⇒ Đưa 3 vào stack và quay về 7.
Tương tự đưa 7, 6, 5, 3, 2, 1 vào stack



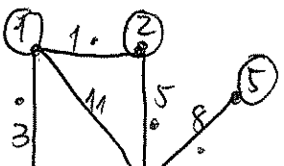
B_{12}: Hoàn tất việc xét tất cả các cạnh của đthị ⇒ KT TT.
Vậy đthị có ctrình Euler:
$1 \to 2 \to 3 \to 5 \to 6 \to 7 \to 3 \to 4 \to 1$.

*



1
2
3
4
2
5
4
3
1

* KRUSKAL:



B_1: Tạo dsách cạnh listEdge = {(1,2), (1,3), (1,4), (2,3), (3,4), (3,5), (3,6)}

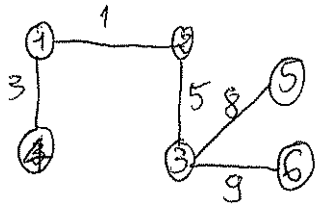
B_2: Sắp xếp listEdge tăng dần theo trọng số:
listEdge = {(1,2), (1,4), (2,3), (4,3), (3,5), (3,6), (1,3)}.

B_3: Tạo $T = \emptyset$

- Lấy từ listEdge ra (1,2), T ∪ (1,2) k° tạo chu trình ⇒ T = {(1,2)}
- Lấy từ listEdge ra (1,4), T ∪ (1,4) k° tạo chu trình ⇒ T = {(1,2), (1,4)}
- -nt- (2,3), T ∪ (2,3) -nt- ⇒ T = {(1,2), (1,4), (2,3)}.
- -nt- (4,3) T ∪ (4,3) tạo ctrình
- -nt- (3,5), T ∪ (3,5) k° tạo ctrình ⇒ T = {(1,2), (1,4), (2,3), (3,5)}
- -nt- (3,6), T ∪ (3,6) -nt- ⇒ T = {(1,2), (1,4), (2,3), (3,5), (3,6)}

B_4: Lúc này, cây có n-1 cạnh ⇒ Dừng TT.

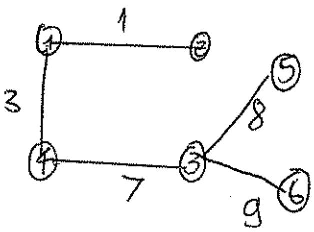
KL: Vậy CKNN là:



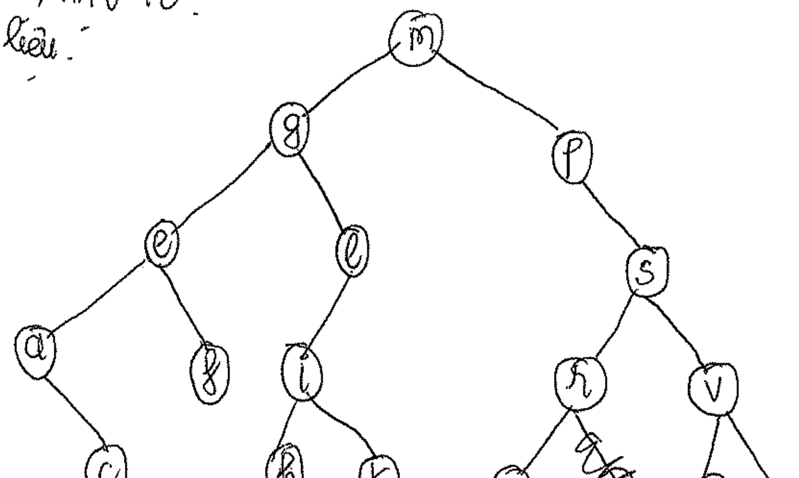
Tổng trọng số của CKNN:
3+1+5+8+9 = 26.

* PRIM:

Vòng lặp	1	2	3	4	5	6	V_T	E_T
Khởi tạo	[1,0]	*[1,1]	[1,11]	[1,3]	∞	∞	1	∅
1	–	–	[2,5]	*[1,3]	∞	∞	1, 2	(1,2)
2	–	–	*[4,7]	–	∞	∞	1,2,4	(1,2),(2,4)
3	–	–	–	–	*[3,8]	[3,9]	1,2,4,3	(1,2),(2,4),(4,3).
4	–	–	–	–	–	*[3,9]	1,2,4,3,5	(1,2),(2,4),(4,3),(3,5).
5	–	–	–	–	–	–	1,2,4,3,5,6	(1,2),(2,4),(4,3),(3,5),(5,6).

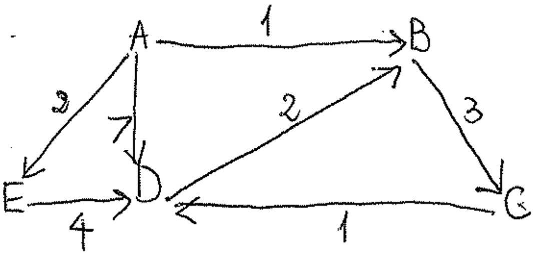


* TIỀN, TRUNG, HẬU TỐ:
Vd/73 tài liệu.



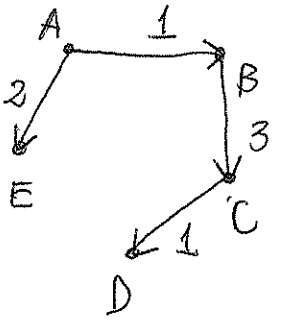
- LNR: a c e f g h i k l m p q r s t v x
- NLR: ~~m g e~~ m g e a c f l i h k p s r q v t x
- LRN: c a f e h k i l g q r t x v s p m

* DIJKSTRA:

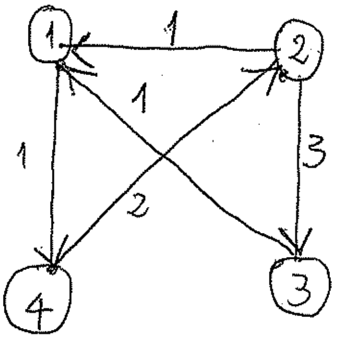


Vòng lặp	A	B	C	D	E
Khởi tạo	[0, A]	*[1,A]	[∞,A]	[7,A]	[2,A]
1	–	–	[4, B]	[7,A]	*[2,A]
2	–	–	*[4, B]	[6,E]	–
3	–	–	–	*[5,C]	–
4	–	–	–	–	–

Vậy đđi ngắn nhất từ A đến các đ^{2} còn lại là:



* FLOYD:



B_1 : Khởi tạo:

	1	2	3	4
1				1,4
2	1,1		3,3	
3	1,1			
4		2,2		

B_2 : Chọn đ^{2} 1 làm trung gian:

	1	2	3	4
1				1,4
2	1,1		3,3	2,1
3	1,1			2,1
4				

- $L[2,3] = L[2,1] + L[1,3] = \infty$
- $L[2,4] = L[2,1] + L[1,4] = 2$
- $L[3,4] = L[3,1] + L[1,4] =$
-