https://boazturya.github.io/cs435ALGO/BOAZ_LAB11A.pdf

LAB 11 A

PROBLEM 1 Must every dense graph be connected? Prove

No, not every dense graph is connected.

Proof

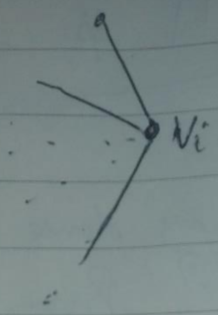
Consider a complete graph having n vertices such that

$$V = \{v_1, v_2, v_3 \ldots v_n\}$$

Since it is complete, the number of edges $m = \frac{n(n-1)}{2}$.

Each vertice, v_i is connected to every other vertice in the graph. Therefore each vertice v_i has $(n-1)$ edges connected to it

Graph



If we remove the vertice v_i from the graph, ~~and~~ the remaining graph has

$$m' = \frac{n(n-1)}{2} - (n-1) \text{ edges}$$

$$= \frac{(n-1)(n-2)}{2}$$

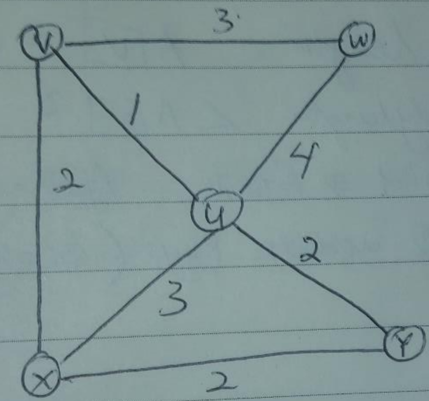
$$m' = \frac{n(n-1)-2n+2}{2} = \frac{n^2-n-2n+2}{2}$$

$$m' = \frac{n^2-3n+2}{2}$$

Therefore consider the graph ~~has~~ G' having $(n-1)$ vertices together with a separated ~~ots~~ component having the the vertice v_i. The total number of edges $m = \frac{n^2-3n+2}{2}$. This is still $\Theta(n^2)$ ~~but the~~ but the graph is disconnected.

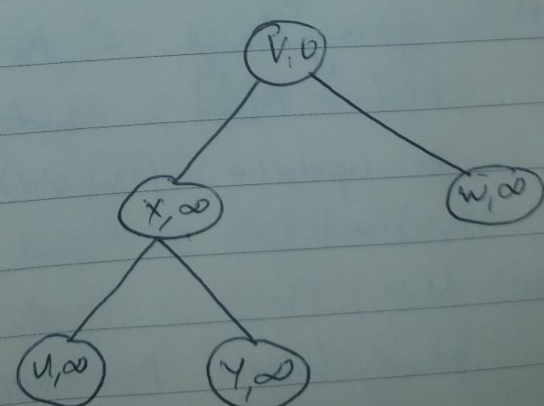
PROBLEM 2

a) The length of the shortest path from v to Y is 3.



b)

	A
A[V]	0
A[X]	∞
A[W]	∞
A[U]	∞
A[Y]	∞



$X = \{ \}$

$M = \{ v, \cancel{X}, w, u, Y \}$

~~Ste~~

Step 1 remove min $(v, 0)$.

vertices adjacent to v: x, w, u

Process x

greeddy length = $A[v] + wt(v,x) = 2$

greedylength $< A[x]$? Yes

old $\Rightarrow A[x]$ $A[x] = 2$.

Q.update key $((x, old), (x, 2))$.

Process w

greedy length = $A[v] + wt(v, w) = 3$.

greedy length $< A[w]$? Yes.

old = $A[w]$, new $A[w] = 3$

Q. update $((w, old), (w, 3))$

Process u.

greedy length = $A(u) + wt(v,u) = 1$

greedy length $< A[u]$? Yes

old = $A[u]$, new $A[u] = 1$.

Mo Tu We Th Fr Sa Su

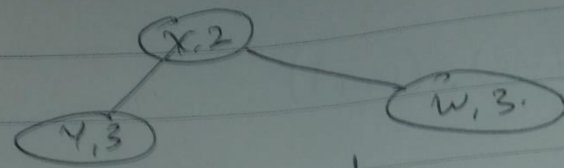
Date / /

Q. update ((u, old), (u, 1))

Step 2 remove min (u).
M. remove u.
vertices adjacent to u: x, w, y.

Process x
greedy length = A[u] + wt (u,x) = 1+3 = 4.
greedy length < A[x]? No.

Process w
greedy length = A[u] + wt(u,w) = 1+4 = 5
greedy length < A(w)? No.
Process y.
greedy length = A[u] + wt(u,y) = 1+2 = 3.
greedy length < A[y]? Yes.
old = A[y] , A[y] = 3.
Q. update key ((y, old), (y, 3))



A	
A[V]	0
A[X]	2
A[W]	3
A[U]	1
A[Y]	3

$X = \{V, U\}$

$M = \{X, Y, W\}$

~~Sol~~ Step 3: remove min (X)

M.remove(X).

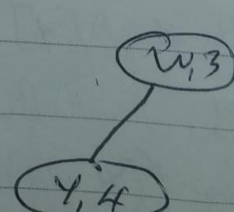
vertices adjacent to X : Y

Process Y:

greedylen = A[X] + wt(X, Y) = 2+2
= 4.

Greedy len < A[Y] ? No

A	
A[V]	0
A[X]	2
A[W]	3
A[U]	1
A(Y)	3



$X = \{V, U, X\}$

$M = \{Y, W\}$

Memo No.
Date / /

Step 4! remove ~~[illegible]~~ min (w).
M. remove (w).
vertices adjacent to w : —
X = {v, u, x, w}
M = {Y}

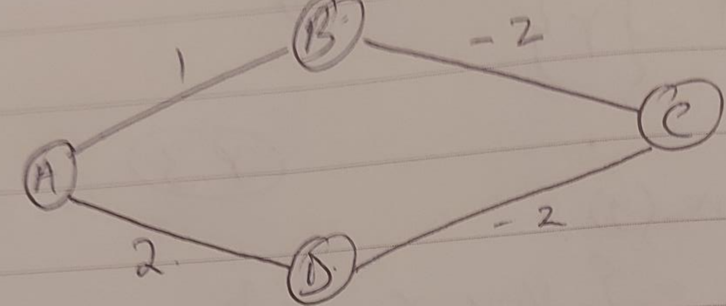
(Y, 3)

remove (y)
X = { v, u, x, w, y}
M = { }.

A	
A[V]	0
A[X]	2
A[W]	3
A[U]	1
A[Y]	3

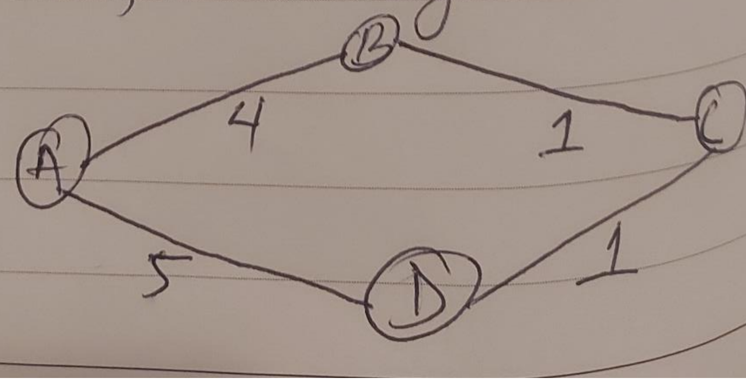
PROBLEM 3

a) Shortest path from A to C.



There is no algorithm that truly computes the shortest path ~~og~~ on undirected graphs with negative edge weights.

We can add a uniform 3 to all the edges and work it out using Dijstra's algorith



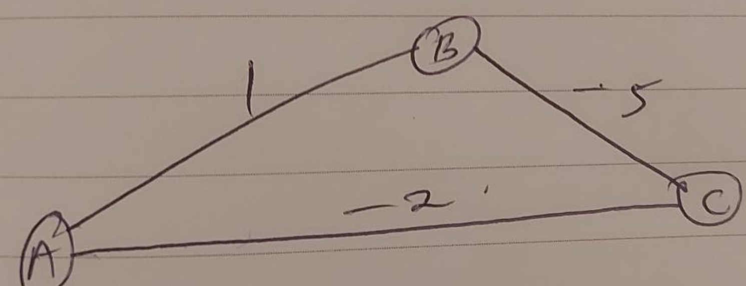
The shortes path from A to C is ABC ⇒ ~~S~~ which is 5.

5 - 3 - 3 = -1

However by continuing the path to ABCBC, path is -5

This kind of path repitition results is ~~graodually~~ ~~conti~~ continuously reducing ~~pta~~ path length.

b)



1) it will compute $A[C] = -2$ because of the greedy length choice

b) ii) Dynamic programming

$D[A] = 0$

$D[B] = \min\{P[\boxed{x}] + wt(x,v) \mid (x,v) \in E\}$

$= D[A] + wt(A,B)$

$= 0+1 = 1$

$D[C] = \min\begin{cases} D[\boxed{A}] + wt(A,C) = -2 \\ D[\boxed{B}] + wt(B,C) = 1-5 = -4 \end{cases}$

$D[C] = -4$

which is Correct