Dijkstras algorithm

Heap

s → ∞, 0

a → ∞, 7

b → ∞, 15, 13

c → ∞, 1

d → ∞, 14

f → ∞, 31, 27, 21

h → ∞, 18

g → ∞, 16

t → ∞, 27, 24

i → ∞, 23

Parent

S → NIL

C → S

A → S

B → 8, A

F → ∝, ~~B~~, H

D → A

G → D

H → D

T → ~~A~~, I

I → F

Dist

S → 0

C → 1

A → 7

B → 13

D → 14

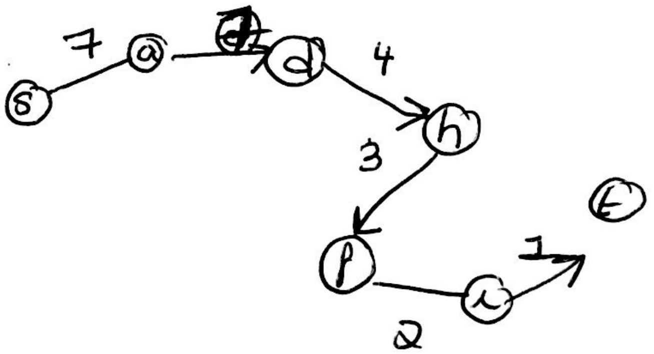
G → 16

H → 18

F → 21

I → 23

T → 24



Floyd Warshall

$D^5(4,6) = \min\{D^4(4,6),\ D^4(4,5) + D^4(5,6)\}$

$= \min\{\infty,\ 7+8\}$

$= \min\{\infty, 15\}$

$= 15$

Print path of (Array, 12, 12, 11, 9)

concat $6 \| P(M, 12, 12, 6, 9)$

$\Downarrow$

$cc\ 12 \| P(M, 12, 12, 12, 9)$

$\Downarrow$

$7 \| P(M, 12, 12, 7, 9)$

$\Downarrow$

$5 \| P(M, 12, 12, 5, 9)$

$\Downarrow$

$2 \| P(n, 12, 12, 2, 9)$.

$\boxed{6 \rightarrow 12 \rightarrow 7 \rightarrow 5 \rightarrow 2}$

Prims

heap

$A \to 0$

$D \to \infty, 5$

$e \to \infty, 5$

$G \to \infty, 5$

$F \to \infty, 5$

$C \to \infty, 30$

$K \to \infty, 20$

$B \to \infty, 10$

$S \to \infty, 10$

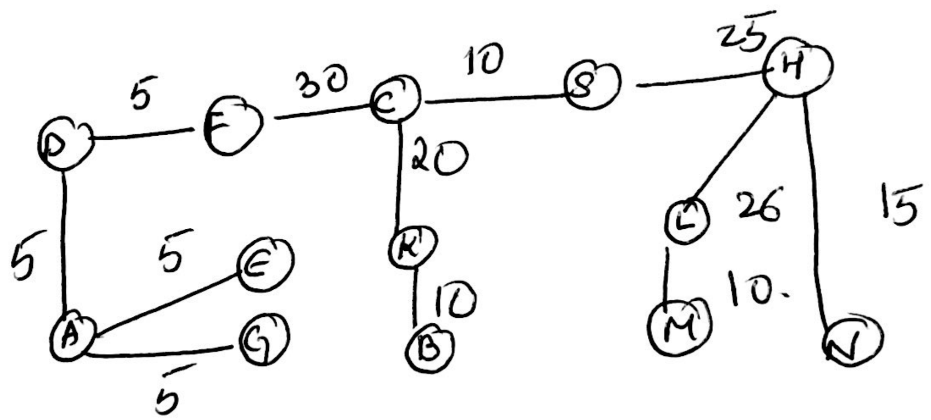
$H \to \infty, 25$

$L \to \infty, 26$

$M \to \infty, 27$

$N \to \infty, 15$

V	Edge
D	AD
E	AE
G	AG
F	DF
C	FC
K	CK
S	CS
H	SH
B	KB
M	~~BM~~ LM
L	HL
N	HN



Kruskals

singleton : {A}, {B}, {C} . . . . {N}

AD - 5 {A, D} ✓

AG - 5 {A, D, G} ✓

AE - 5 {A, D, G, E} ✓

DE - 5 X (both in same set)

DF - 5 {A, D, G, E, F} ✓

EG - 5 X

EF - 5 X

CS - 10 {A, D, G, E, F} {C, S} ✓

BK - 10 {A, D, G, E, F} {C, S} {B, K} ✓

LM - 10 {A, D, G, E, F} {C, S} {B, K} {L, M} ✓

HN - 15 {A, D, G, E, F} {C, S} {B, K} {L, M} {H, N} ✓

CK - 20 {A, D, G, E, F} {C, S, B, K} {L, M} {H, N} ✓

SH - 25 {A, D, G, E, F} {C, S, B, K, H, N} {L, M} ✓

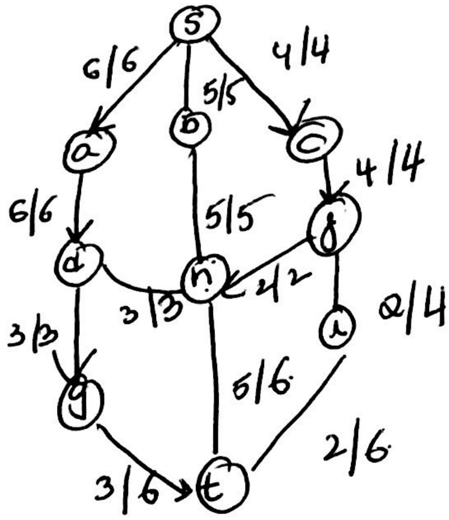
HL - 26 {A, D, G, E, F} {C, S, B, K, H, N, L, M} ✓

BM - 27 X

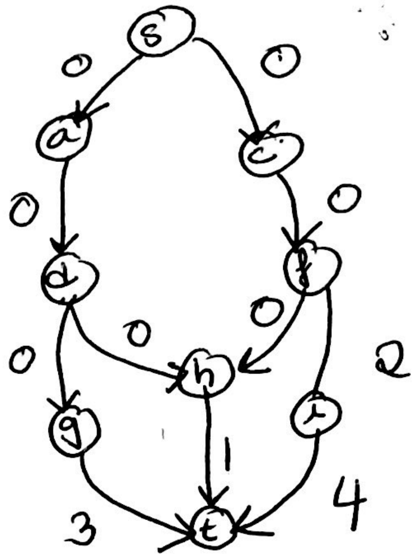
CF - 30 ✓ {A, D, G, E, F, C, S, B, K, H, N, L, M}

MN - 30 X

Ford - Fulkerson



⇒



Path - 1 $\quad s \overset{6}{-} a \overset{6}{-} d \overset{3}{-} g \overset{6}{-} t$

Path 2 $\quad s \overset{3}{-} a \overset{3}{-} d \overset{3}{-} h \overset{6}{-} t$

Path 3 $\quad s \overset{4}{-} c \overset{4}{-} f \overset{2}{-} h \overset{3}{-} t$

Path 4 $\quad s \overset{2}{-} c \overset{2}{-} f \overset{4}{-} i \overset{6}{-} t$

Scanned by CamScanner

5. (10 points) Please circle, put a big check-mark, or say "YES" in the correct box (no need to explain, there is exactly one correct answer for each question.)

1)	If a decision problem X is in $\mathcal{NPC}$, then the worst case time complexity of every published deterministic algorithm that affirms (i.e. solves) X is "at least polynomial", i.e. $T^W(n) = \Omega(n^k)$ for every integer k.	known to be TRUE	known to be FALSE	open question
2)	If a decision problem X is in $\mathcal{NPC}$, and if in the future someone publishes a new deterministic algorithm G that affirms (i.e. solves) it and G runs in polynomial time, i.e $T_G^W(n) = \mathcal{O}(n^k)$ for a particular integer k, then this proves that $\mathcal{P} = \mathcal{NP}$.	known to be TRUE	known to be FALSE	open question
3)	If a decision problem X is in $\mathcal{NPC}$, and if in the future someone publishes a new deterministic algorithm G that affirms (i.e. solves) it and G runs in polynomial time, i.e $T_G^W(n) = \mathcal{O}(n^k)$ for a particular integer k, then every $\mathcal{NPC}$ problem can be affirmed (i.e. solved) in polynomial time.	known to be TRUE	known to be FALSE	open question
4)	If a decision problem X is in $\mathcal{NP}$, and if in the future someone publishes a new non-deterministic algorithm G that affirms (i.e. solves) it and G runs in polynomial time, i.e $T_G^W(n) = \mathcal{O}(n^k)$ for a particular integer k, then this proves that $\mathcal{P} = \mathcal{NP}$.	known to be TRUE	known to be FALSE	open question
5)	If a decision problem X is in $\mathcal{NPC}$, and if in the future someone publishes a new non-deterministic algorithm G that affirms (i.e. solves) it and G runs in polynomial time, i.e $T_G^W(n) = \mathcal{O}(n^k)$ for a particular integer k, then this proves that $\mathcal{P} = \mathcal{NPC}$.	known to be TRUE	known to be FALSE	open question
6)	If a decision problem X is in $\mathcal{NPC}$, and if in the future someone publishes a new non-deterministic algorithm G that affirms (i.e. solves) it in polynomial time, i.e $T_G^W(n) = \mathcal{O}(n^k)$ for a particular integer k, then every $\mathcal{NPC}$ problem can be affirmed (i.e. solved) by a non-deterministic algorithm in polynomial time.	known to be TRUE	known to be FALSE	open question
7)	Suppose there is a decision problem $X \in \mathcal{NP}$ and there is a decision problem $Y \in \mathcal{NP}$, and that $X \propto Y$, then this proves that $Y \in \mathcal{NPC}$	known to be TRUE	known to be FALSE	open question
8)	Suppose there is a decision problem $X \in \mathcal{NP}$ and that $X \propto Y$ for all $Y \in \mathcal{NP}$, then this proves that $X \in \mathcal{P}$	known to be TRUE	known to be FALSE	open question
9)	Suppose there is a decision problem $X \in \mathcal{NP}$ and that $X \propto Y$ for all $Y \in \mathcal{NP}$, then this proves that $X \in \mathcal{NPC}$	known to be TRUE	known to be FALSE	open question
10)	Suppose there is a decision problem $Y \in \mathcal{NPC}$ and that $X \propto Y$ for all $X \in \mathcal{NP}$, then this proves that $Y \in \mathcal{NPC}$	known to be TRUE	known to be FALSE	open question