AAI Assignment 2 : 2023113019

Q1(a) $G \equiv (V, E)$

Let edges be sorted & arranged in ascending order

$[e_i = e_{u,v},\ s.t.\ e_i \le e_{i+1}\ \forall i \in [1:|E|]]$ $\{u, v \in V\}$

Using Kruskal's algorithm on G, we get MST T with $(|V|-1) = k$ edges

Thus, we keep choosing e_i unless it creates a cycle. (from Kruskal)

Thus, for some i, e_i is ommitted from $\{MST\ T \equiv (V_T, E_T)\}$ E_T

Let $e_j \le e_{j+1}$ holds because $\{e_j \in E_T\}$ $\forall\ p > 0$

$e_i \le e_{i+1}$ holds & only minimums are chosen at every pt.

Let e_i be E_i, $i \in [1:k]$

$\because$ we know $e_1 < e_2 < \dots < e_{|E|}$

$\Rightarrow ce_1 < ce_2 < \dots < ce_{|E|}$ $(c > 0)$

using Kruskal's we again arrange it taking min. until edge creates cycle.

We also know, $E_1 < E_2 < \dots < E_k$

$\Rightarrow cE_1 \le cE_2 \le \dots \le cE_k$ $(c > 0)$ hence

Since ordering holds, we get same MST T when edges are multiplied by c

$\therefore$ MST remains same.

$\Pi(s,t)$ remains same because of the same argument of ordering $\forall e_i$ remaining same.

Proof by contradiction

Let us assume $\Pi'(s,t) \ne \Pi(s,t)$ is new shortest path

We know, $\Pi(s,t) = \sum e_i$ for some $i \in [1:|E|]$

Let $\Pi'(s,t) = \sum ce_j$ for some $j \in [1:|E|]$ s.t. not all $j = i$

We know, $\sum e_i < \sum e_j$ because $\Pi(s,t)$ was shortest path

$\Rightarrow \sum c_{e_i} < \sum c_{e_i}$

Thus Π' is not shortest path.

$\Pi(s,t)$ $\Pi'(s,t)$

(b) MST Same argument as 1(a) using Kruskal's algorithm

$E_1 < E_2 < \ldots < E_k$ [for MST T]

$\rightarrow E_1 + c < E_2 + c < \ldots < E_k + c$

~~Since ordering holds~~

& also $c_1 + c < e_2 + c < \ldots < e_{|E|} + c$

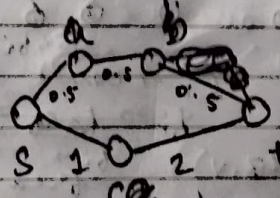
since ordering is the only thing that matter in Kruskals apart from cycles (and no new cycles are created due to edge set remaing constant).

MST ~~remain~~ ordering remains same

$\Rightarrow$ ~~MST~~ T is ~~same~~ still an MST

$\Pi(s,t)$ will not be shortest distance

Counter example

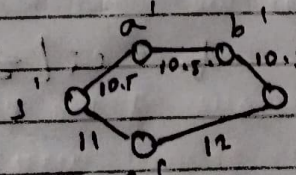


$\Pi(s,t)$ $G = (V, E)$

$= \{s, a, b, t\}; \{e_{s,a}, e_{a,b}, e_{b,t}\}$

1.5 is length

for $c = 10$,



t' length of shortest path = 23

Shortest path here is not ~~$\Pi(s,t)$ here~~

$s' \rightarrow a' \rightarrow b' \rightarrow t'$

but $s' \rightarrow e' \rightarrow t'$ Hence not true

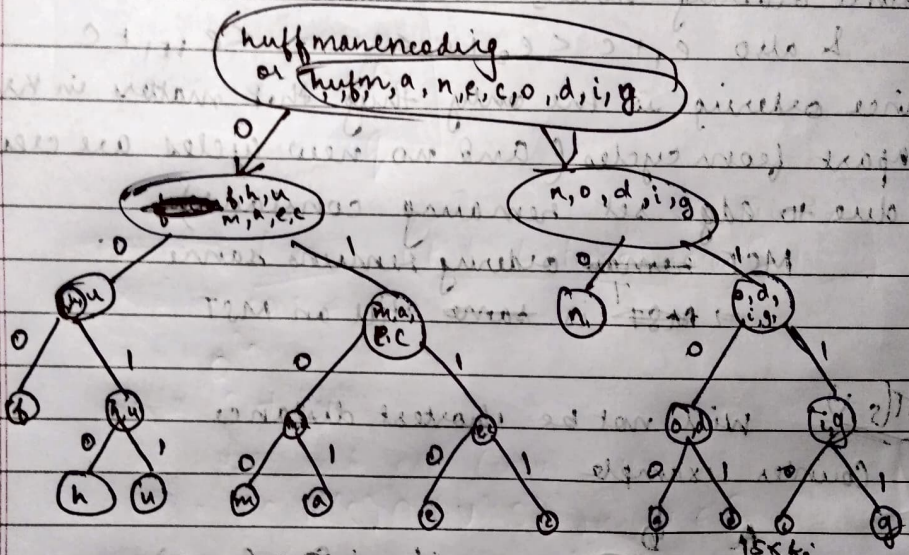
{where $i' = i + c$}

2(a) Huffman Encoding

char	h	u	f	m	a	n	e	c	o	d	i	g
count	1	1	2	1	1	3	1	1	1	1	1	1

freq = count / Σcount Σcount = 15 here

Combining the least frequency characters, we create the final tree:



Total Bitlength = $\Sigma f_i d_i$

$= 10 \times (1 \times 4) + 1 \times (2 \times 3) + 1 \times (3 \times 2)$

$= \boxed{52}$ Ans

(b) Avg bit length = $\Sigma f_i d_i$ $f_i = c_i / \Sigma c_i$

$= \frac{\Sigma c_i d_i}{\Sigma c_i}$

$= \boxed{\frac{52}{15}}$ Ans ≈ 3.46

3: We have to handle water supply problem, which looks a lot like the MST problem, with an additional 'well' part. We make the following observations that are conditions.

1. Water supply must have at least 1 well (source of water)
2. Every house must receive water either by well or pipe.
3. Cost must be minimized.

We can hence model the problem with H_i $i \in \mathbb{N}$, $i \in [1:N]$ denoting the N Houses. To this set of vertices (of a graph we are about to create), we add vertex H_0 (which would denote well). Let $c_{i,j}, c_{0,k}$ $\forall i,j,k \in \mathbb{N}$, $i,j,k \leq N$

Assuming $c_{i,j}, w_k$ are all finite,

we know $e_{u,v}$ exist $\forall u,v \in V$.

$\therefore G \equiv (V_G, E_G)$ is a complete graph.

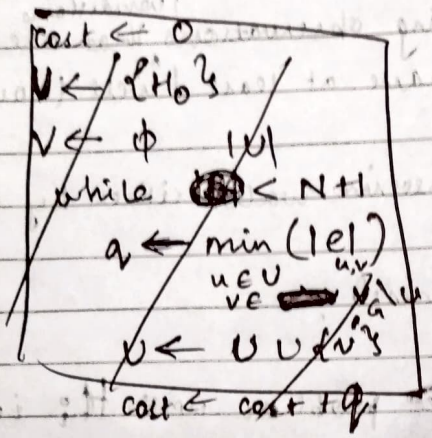
Our problem reduces to finding an MST for this graph, because that ensures: 1. (V_G contains H_0) & 2. (V_G contains H_i), 3. MST is minimum cost.

Formally,

$E = \{ e_{ij} \mid e_{ij}$ is edge b/w H_i & H_j $(i \neq j)$ & $wt(e_{0,i}) = wt(w_i)$ & $wt(e_{i,j}) = c_{i,j}$ $\forall i \neq j, i \in \mathbb{N} \}$

PTO

Now, we can use any MST algorithm, say Prim's to find MST:



Algo (Ref. Lecture notes for Prim's & similar)

cost ← 0

S = {H0}

while S ≠ |V|

Pick u, v and edge (u, v) s.t that

- v ∈ N(S) ∩ (V \ S) and u ∈ S
- wt (u, u) ≤ wt (u', v') ∀ u' ∈ S, v' ∈ N(S) ∩ (V \ S)

S ← S ∪ {v}

cost ← cost + wt (edge (u, v))

return cost

This path is not stored because we want only the cost, improving space complexity.

This algorithm is true because of the known property that Prim's algorithm is an optimal way to find MST (derived from the cut property) for non-negative edge weighted graphs [$c_{ij} > 0$ & $w_k > 0$]

Q Let $G \equiv (V, E)$, $n \to |V|$

Algo (Ref web.stanford.edu)

$d^0[s] \leftarrow 0$

For all $u \in V \setminus \{s\}$

$d^0[u] \leftarrow \infty$

~~For k = 1,~~

$d^k[v] = \text{None} \quad \forall v \in V, k \in [1 : n-1]$

for $k = 1, \ldots, n-1$ do

$d^k[v] \leftarrow d^{(k-1)}[v]$

for $e_{(u,v)} \in E$

$d^k[v] \leftarrow \min\{ d^k[v], d^{k-1}[u] + wt.(e_{(u,v)}) \}$

$d^{k-1}[v] \doteq \text{None} \quad \forall v \in V$

visit $\leftarrow \{\}$

~~return~~

for $e_{(u,v)} \in E$

if $d^{n-1}[v] \neq \infty$ and $d^{n-1}[v] + wt.(e_{(u,v)}) < d^{n-1}[v]$

$d[v] = -\infty$

~~$d[v] = -\infty$~~ for $e_{u,v'} \forall u \in N(1)$, $v' \in V \cap$ visit

$d[v'] = -\infty$, visit $\leftarrow$ visit $\cup v'$

~~return d~~

return $d^{n-1}[v] \quad \forall v \in V$

"This uses a Bellman - Ford sort of implementation & uses the fact that by n^{th} iteration, if shortest path

is not found, ~~it~~ it must have a -ve cycle, hence all ~~is equal to~~ vertices ~~in that~~ reachable from that path are $-\infty$