Lecture 07

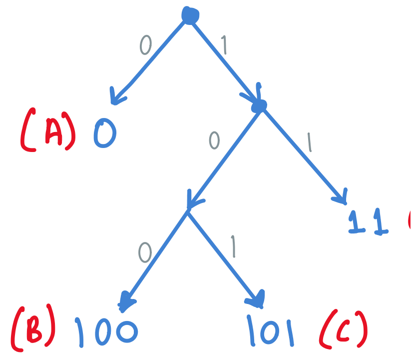
Tuesday, 24 August 2021 9:06 AM

Huffmann Encoding

	FREQ
A	45
B	9
C	11
D	35

$\rightarrow$

	CODES
A	0
B	100
C	101
D	11



Length of $\overset{\text{Encoded}}{\text{msg}}$ = 45(1) + 9 (3) + 11

Observation: If $a_1 .. a_n$ satisfy $f_1 \geq f_2 \geq \dots \geq f_n$. Then, there
(last class)
opt tree T where a_n, a_{n-1} are siblin

Prefix encoding

Sketch of Algo:

① Replace two letters with least frequency, i.e. a_n, a_{n-1} by single new symbol $\tilde{a}$

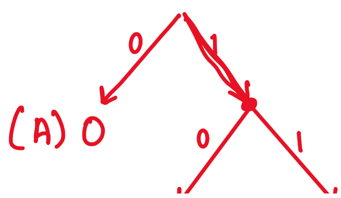
② So, $freq(\tilde{a}) = \tilde{f} := f_n + f_{n-1}$

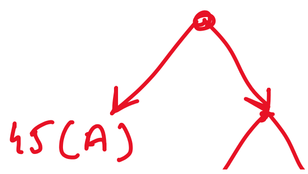
③ Solve $\tilde{F} = (F \cup \tilde{f}) \setminus \{f_{n-1}, f_n\}$, and

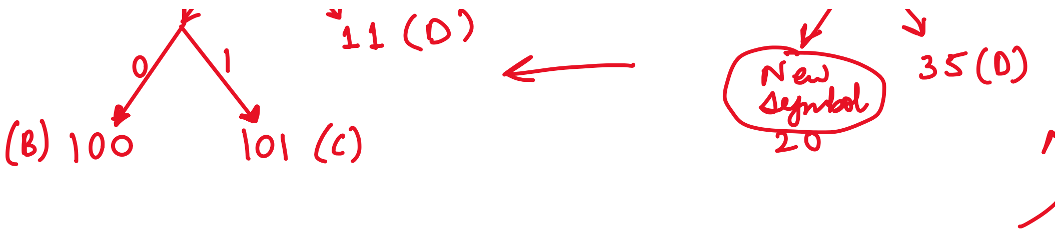
eliminating

④ If $\tilde{a}$ is node for $\tilde{f}$, then add children a_n

Huffmann.







F $(45, \underline{9}, \underline{11}, 35) \longrightarrow F^* = (45, 20, 35)$

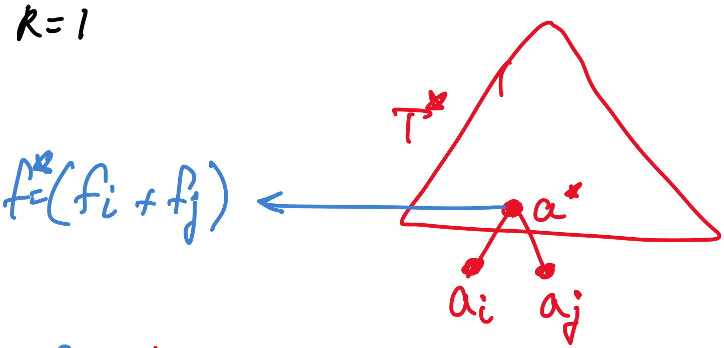
General Theorem: Let $F = (f_1, \ldots, f_n)$ represent frequency of sy
and $\underline{i, j}$ be such that $\exists$ an opt tree in which
Then, problem $F^* = (F \setminus \{f_i, f_j\}) \cup \{f^*\}$ where
satisfy $\quad opt(F) = f_i + f_j + opt(F^*)$

Proof:

(i) $opt(F) \leq opt(F^*) + f_i + f_j$

Let T^* be opt tree of F^*, and
let a^* be node with freq $f^* = f_i + f_j$
Create a tree T with n leaf nodes from
T^*, by just adding children a_i, a_j to a^*

$$\sum_{k=1}^{n} f_k \cdot dept(a_k, T) = opt(F^*) + \underbrace{f_i + f_j}$$



$f_i \quad depth(a_i) = \underline{1} + depth(a^*)$
$f_j \quad depth(a_j) = \underline{1} + depth(a^*)$

(ii) opt(F

Use Fact:

in

T

H.W. Soln

Let a^* be
be tree ob
a_i, a_j.
Then $\tilde{T}$
$opt(F^*) \leq$

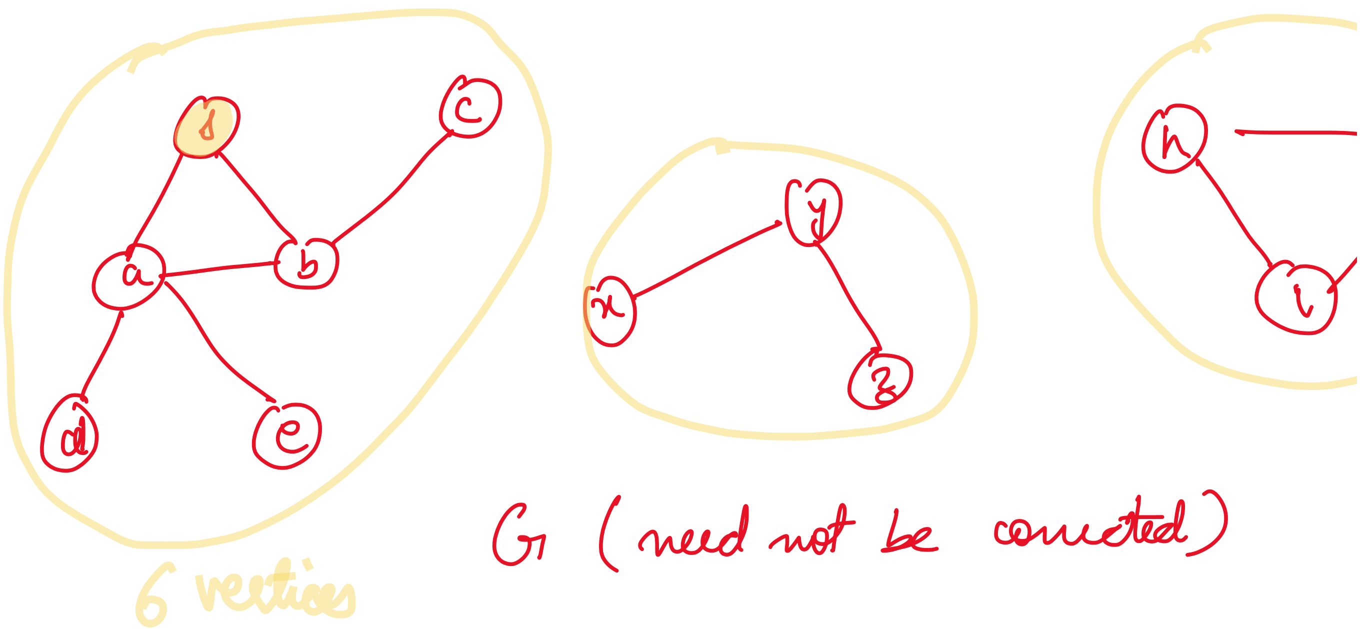
$\Rightarrow opt(F) \leq sol^n\text{-}size(T) = opt(F^*) + f_i + f_j$

$\Rightarrow opt($

$\circledast$ BFS / DFS / shortest-path algo in weighted

BFS — shortest path tree

DFS — connectivity



G (need not be connected)

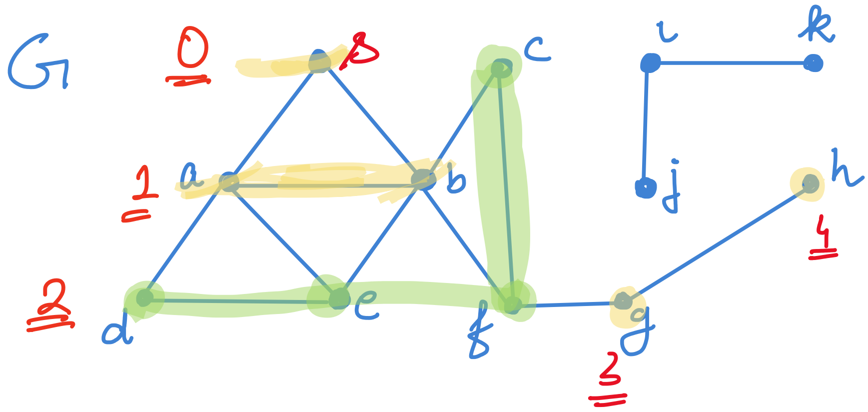
$L_0 = \{s\}$

$L_1 =$ All neighbors of s.

$L_2 = \{x \mid x$ is a neighbor of a vertex

$\vdots$

$L_i = \{x \mid x$ is a neighbor of a vertex



$L_0 = \{s$

$L_1 = \{a,$

$L_2 = \{c,$

$L_3 = \{$

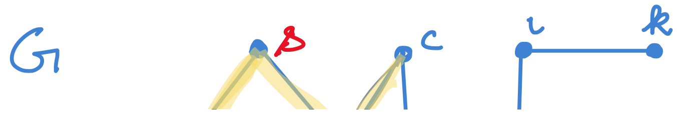
$L_4 = \{h$

Claim: If G is unweighted, then

$$L_i = \{ x \mid dist(s, x, G) = i \}$$

Proof: H.W. (by induction).

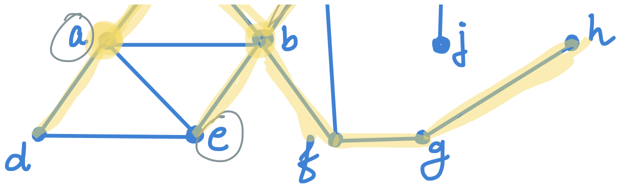
Example:



Question:

Can you

$S_1 S_2 -$



$S_1 \quad S_2$

flag = visit

$S_1 = \boxed{s}$

$S_2 = \boxed{a}$

$S_1 = \boxed{c}$

$S_2 = \boxed{g}$

code for this? ←

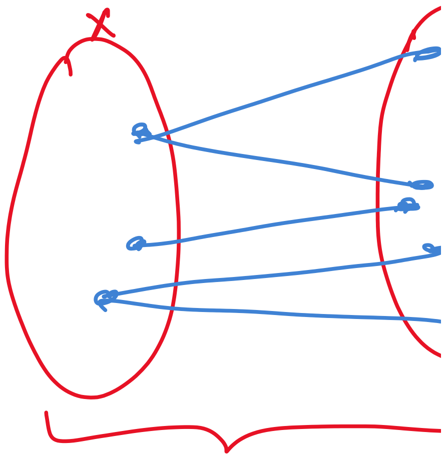
$V = \{1, \ldots, n\}$

⊛ stacks with integer entries is sufficent

⊛ Application of BFS Tree.

Bipartite graph (defⁿ):

An undirected graph $G=(V,E)$ for which there is a partition (X,Y) of V satisfying $E \subseteq X \times Y$.



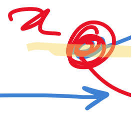
$V(G) = X$

each edge (

Ques 1) If T is a BFS tree of

for each edge (x,y), leve

Then, prove that G is Bipartit

not possible $\longrightarrow$ 

Ques 2 : If G is bipartite, then G ha

Ques 3 : Use Q1, Q2 to obtain $O(m$
if a given graph is bip