COL 351: Analysis and Design of Algorithms

Lecture 23

Q1(a): Design an algorithm that verifies whether a given undirected graph (not necessarily connected) with n vertices and m edges is acyclic or not in $O(n)$ time.

Perform DFS traversal of G and stop if

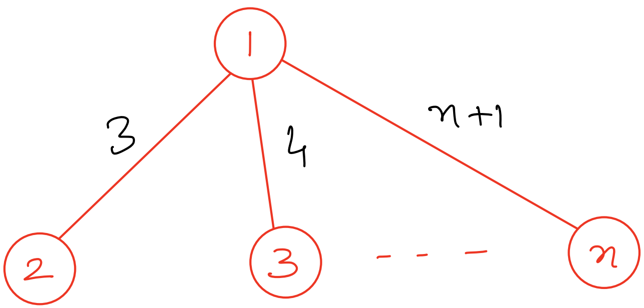
i - all vertices are visited, or

ii - you find a <u>back edge</u> / a vertex is visited "twice"

DFS

1 extra edge.

Q1(b): Let $G = K_n$ be a complete graph on n vertices where for any distinct $i, j \in [1,n]$ $weight(i,j) = i + j$. Find an MST of G.



$wt(MST) = 3 + 4 + \cdots + n + n+1$

Reasoning:

$i \longrightarrow$ wt is $(i+1)$

All other edges incident to "i" have wt $> (i+1)$.

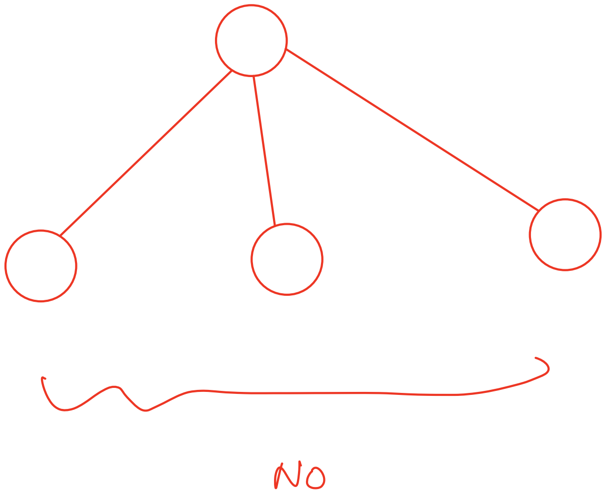
Q1(b): Let $G = K_n$ be a complete graph on n vertices where for any distinct $i, j \in [1,n]$ $weight(i,j) = |i - j|$. Find an MST of G.

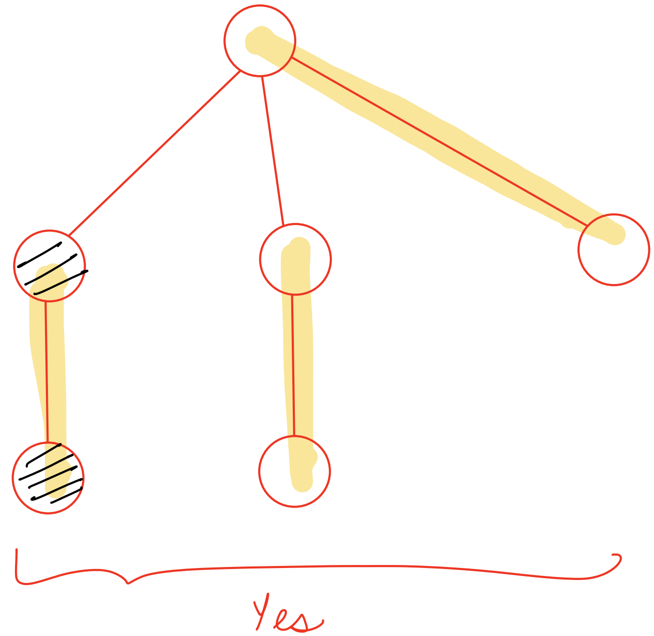
$wt(MST) = 1 \times (n-1) = n-1$

Reasoning: We choose $n-1$ edges of wt 1.

All other edges have wt > 1.

Q2: Design a linear-time algorithm that takes as input a forest F on n vertices and determines whether it has a perfect matching: a set of edges that touches each node exactly once.

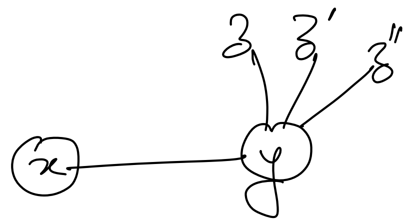




Q2: Design a linear-time algorithm that takes as input a forest F on n vertices and determines whether it has a perfect matching: a set of edges that touches each node exactly once.

$X_0 \leftarrow$ Vertices of degree 0 in F

$X_1 \leftarrow$ Vertices of degree 1 in F



While (V(F) is non-empty)

{

If $|X_0| > 0$ Return False

Else

$x \leftarrow$ any vertex of deg 1.

$y \leftarrow$ neighbor of x.

Remove x, y from $V(F)$

For $z \in N(y) \setminus x$: $\deg(z) = \deg(z) - 1$.

If $\deg(z) = 0$: add z to X_0

— " — $= 1$: add z to X_1

}

Return True.

Q3: Given is a sequence $X = (x_1, \ldots, x_n)$. Design an $O(n^2)$ time algorithm to find the minimum number of characters that needs to be inserted in X to get a palindrome.

ans (L A B E L) = 2

L A E B E A L

ans (A B E) = 2

Q3: Given is a sequence $X = (x_1, \ldots, x_n)$. Design an $O(n^2)$ time algorithm to find the minimum number of characters that needs to be inserted in X to get a palindrome.

$OPT(i,j)$ = Sol^n for substring $X[i,j]$

$OPT(i,j) = -\infty \quad j < i$

$OPT(i,i) = 0 \quad , \forall i$

$$OPT(i,j) = \begin{cases} OPT(i+1, j-1) & x_i = x_j \\ \text{MIN}\begin{cases} 1 + OPT(i, j-1) \\ 1 + OPT(i+1, j) \end{cases} & \text{o/w.} \end{cases}$$

$O(1)$ time for single (i,j)

Imp Ques: In which order should you visit vertex pairs?

Need to handle (i,j) pairs in increasing order of value of $|i-j|$.

Q4: Let G be a DAG with vertex-set $\{1, 2, \ldots, n\}$ and topological ordering $(1, 2, \ldots, n)$. For each pair (x, y), let $N(x, y)$ denote the number of distinct paths from x to y in G.

Let $\cong$ be an equivalence relation on vertex-pairs of G, such that $(a, b) \cong (c, d)$ iff $N(a, b) = N(c, d)$.

(a) Provide a recursive relation to compute $N(x, y)$ from $N(x_1, y), \ldots, N(x_t, y)$, where $x_1, \ldots, x_t$ are out-neighbours of x in G.

$\mathcal{P}$ = Collection of ALL $(x \rightsquigarrow y)$ paths in G

$\mathcal{P}_i$ = Collection of ALL $(x \xrightarrow{\text{edge}} x_i \overset{\text{path}}{\rightsquigarrow} y)$ paths in G (successor of x is x_i)

$$\boxed{N(x,y) = \sum_{i=1}^{t} N(x_i, y)}$$

Here we assume $N(x,x) = 1$

If some one has defined $N(x,x) = 0$, then eqns will change

Q 4 (a) Part 2 : Show how to compute $N(x,y)$, for all pairs (x,y) in G_0

- Initialize $N(x,x) = 1$, $\forall x$
- Set $N(x,y) = 0$ whenever $y < x$

- Finally use the recursive relation to compute $N(x,y)$.

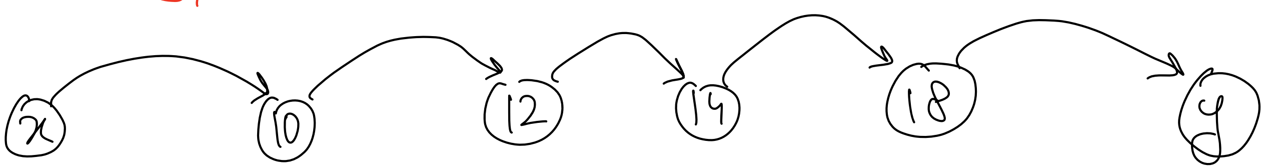
Q4: Let G be a DAG with vertex-set $\{1,2,\ldots,n\}$ and topological ordering $(1,2,\ldots,n)$. For each pair (x,y), let $N(x,y)$ denote the number of distinct paths from x to y in G.

Let $\cong$ be an equivalence relation on vertex-pairs of G, such that $(a,b) \cong (c,d)$ iff $N(a,b) = N(c,d)$.

(b) Argue that for any (x,y), $N(x,y) \leq 2^n$. Next show that for any two pairs (a,b), (c,d) the no. of prime factors of $|N(a,b) - N(c,d)|$ is at most n.

Assume $x < y$.

Each $x \rightsquigarrow y$ path in G $\xrightarrow[\text{to}]{\text{corresponds}}$ subset of $\{x+1, \ldots, y-1\}$



$\{10, 12, 14, 18\}$ a subset of $[x+1, y-1]$

$2^n \leftarrow$ possible # of subsets

4(b) Part 2

Let $X = |N(a,b) - N(c,d)| \leq 2^n$

$X = p_1 \; p_2 \; p_3 \; \cdots \; p_\alpha$ $\quad (p_i \geq 2)$

$\Rightarrow \alpha \leq n$

So, no of prime factors $\leq n$

Q4: Let G be a DAG with vertex-set $\{1,2,\ldots,n\}$ and topological ordering $(1,2,\ldots,n)$. For each pair (x,y), let $N(x,y)$ denote the number of distinct paths from x to y in G.

Let $\cong$ be an equivalence relation on vertex-pairs of G, such that $(a,b) \cong (c,d)$ iff $N(a,b) = N(c,d)$.

(c) Design an $O(mn)$ time algorithm that computes with probability $(1-1/n)$ the equivalence classes under equivalence relation $\cong$.

Let p = random prime in range $[2, n^7]$.

Let $H : x \longrightarrow (x) \bmod p$ be a hash function.

<u>Note:</u> If $x_1 \ldots x_t$ are out-neighbors of x, then

$$\underbrace{H(N(x,y))}_{O(t)} = H(N(x_1,y)) + \cdots + H(N(x_t,y)) \bmod p$$

- $O(m)$ time for single y
- $O(mn)$ $\forall y$

We say $(a,b) \cong (c,d)$ iff $H(N(a,b)) = H(N(c,d))$

Take 2 pairs (a,b), (c,d) with $N(a,b) \neq N(c,d)$

$$\text{Prob}\big(H(N(a,b)) = H(N(c,d))\big) =$$

$$\text{Prob}\big(p \text{ divides } N(a,b) - N(c,d)\big) \leq \frac{n}{\Theta(n^7/\log n^7)} \leq \frac{1}{n^5}$$

By union bound

$$\text{Prob of error in entire algo} = \frac{\#\text{ of choices for }(a,b),(c,d)}{n^5} \leq \frac{n^4}{n^5} \leq$$

$$\leq 1/n$$

Fact: For number of size n^c

we can $+$, $-$, $*$, mod.

in $O(c)$ time

WORD RAM Model

working with n, n^2

bus size $= O(\log_2 n)$

64 bits

$\downarrow$

$O(1)$ time

Fix a "y"

$$H(N(x,y)) = \left(H(N(x_1,y)) + \dots + H(N(x_t,y)) \right) \bmod p.$$

Fix a y

$$\text{Time} = O\left(\sum_x |\text{out-neighbour}(x)|\right) = O(m)$$

$\forall y$ $\quad O(m \cdot n)$