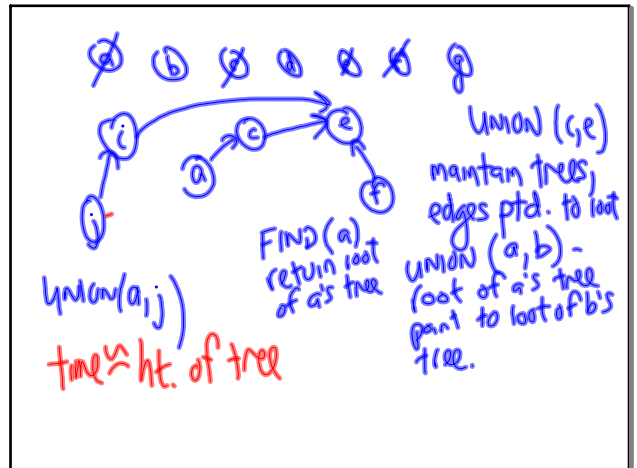
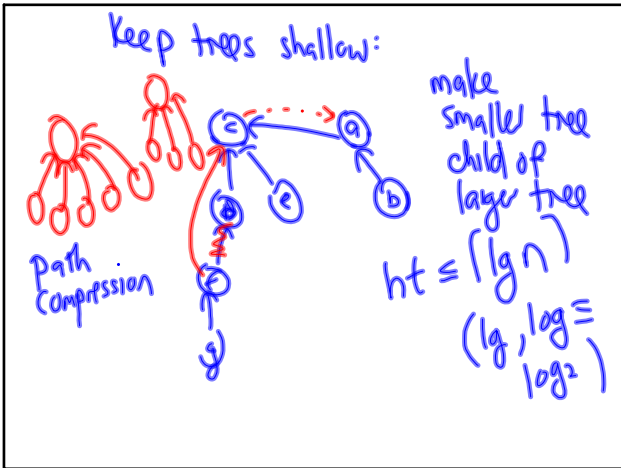


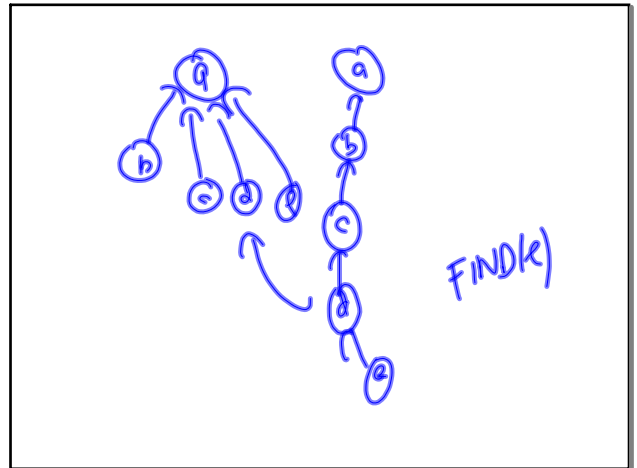
Jan 26-10:59 AM



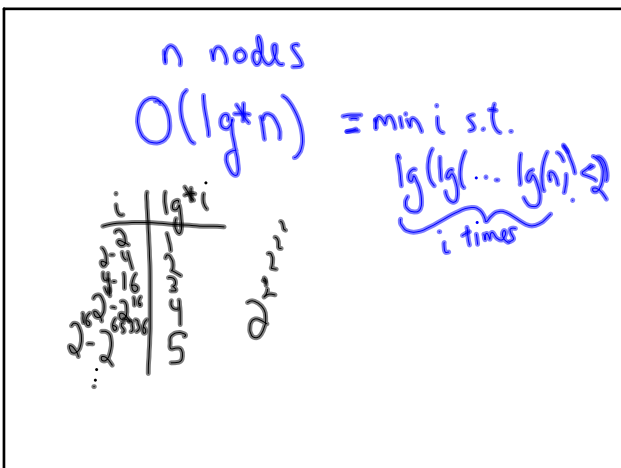
Jan 26-11:04 AM



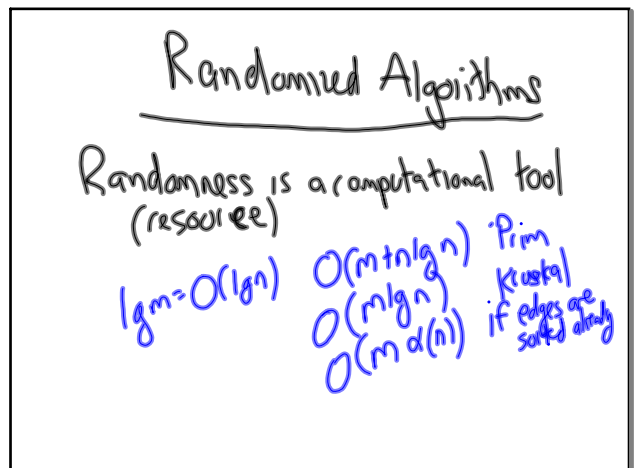
Jan 26-11:10 AM



Jan 26-11:17 AM



Jan 26-11:20 AM



Jan 26-11:31 AM

$$A_k(j) = \begin{cases} j+1 & \text{if } k=0 \\ A_{k-1}^{(j+1)}(j) & \text{if } k \geq 1 \end{cases}$$

$$\alpha(n) = \min\{k : A_k(1) \geq n\}$$

$$A_0(j) = j+1$$

$$A_1(j) = A_0^{(j+1)}(j) = 2j+1$$

$$A_2(j) = A_1^{(j+1)}(j) = 2(2 \cdots (2j+1) \cdots + 1) + 1$$

Jan 26-11:35 AM

$$E(\text{Time}(I)) = E_{\text{Conflops}} [\text{Alg}(I, \text{conflops})]$$

$$\max_I E(\text{Time}(I))$$

$$Pr(\text{Time} \geq 2) < \frac{1}{2^n}$$

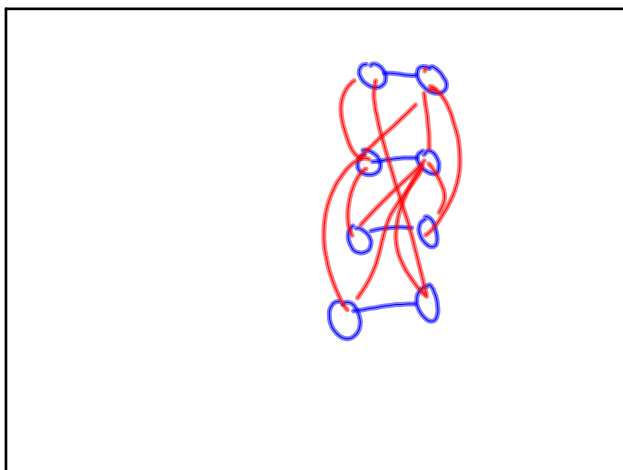
Jan 26-11:35 AM

wt. vertices Example
 Choose $\geq \frac{n}{2}$ edges
 $G/(v,w)$
 remove v,w new node vw incident to all nodes that were incident to v or w

Jan 26-11:53 AM

Each iteration $\leq \frac{n}{2}$
 $\Rightarrow \leq \lg n$ iterations
 implement iteration in $O(n)$ time.
 $O(n \lg n)$ time.
 $n + \frac{n}{2} + \frac{n}{4} + \dots \leq 2n$
 $T(n) = T(\frac{n}{2}) + O(n) \Rightarrow O(n)$

Jan 26-11:53 AM



Jan 26-11:57 AM

Eliminating Edges $w_F(s,x) = \infty$

- The heaviest edge on any cycle is not in the MST.
- Let F be a forest.
- Let $w_F(u,v)$ be the maximum weight of an edge on the path from u to v in F (or ∞ if the path does not exist).
- Edge (u,v) is F -heavy if $w(u,v) > w_F(u,v)$ and F -light otherwise.

Claim: Let F be any forest, let (u,v) be any edge. If (u,v) is F -heavy, then (u,v) is not in the MST.

$w_F(u,v) = 10$
 $w(u,v) > w_F(u,v)$
 uv is F -heavy
 xy is F -light

Jan 26-12:09 PM

Ideas

- It is good to eliminate **F-heavy** edges
- The MST T would let us eliminate all non-MST edges.
- What can an F that is not an MST eliminate

Fact to accept without proof: Given G and a forest F , we can eliminate all F -heavy edges in $O(n+m)$ time (spanning tree verification).

Jan 26-12:13 PM

Algorithm $MST(G)$

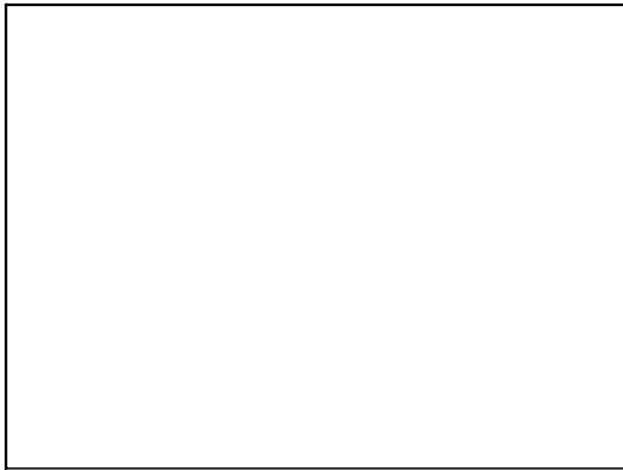
1. Run 3 Brvka phases to get G' . Let C be the contracted edges.
2. Let G'' be G' with each edge included with prob. $1/2$.
3. Recursively compute $F'' = MST(G'')$.
4. Identify the F'' -heavy edges in G' . Delete them to obtain G'' .
5. Recursively compute $F'' = MST(G'')$.
6. Return $F'' \cup C$.

Note: The recursion bottoms out on a graph with $O(1)$ nodes.

Key Lemma: Let H be a subgraph of G where each edge is included with probability p . Let F be a Minimum Spanning Forest of H . Then the expected number of F -light edges in G is at most n/p .

Recurrence $T(n, m) \leq O(n+m) + T(n/8, m/2) + T(n/8, n/4) = O(n+m)$

Jan 26-12:19 PM



Jan 26-11:58 AM