CS218 ASSIGNMENT 3

due Thursday, February 21, at 5PM

Individual assignment: Problems 1,2

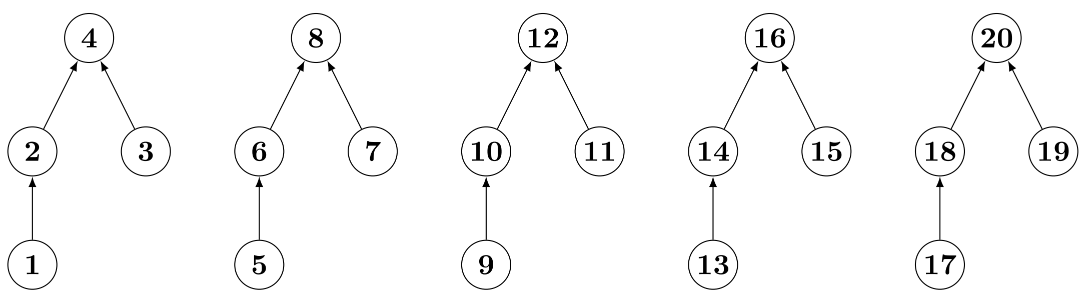
Group assignment: Problems 1,2,3

Problem 1: Suppose that we are using disjoin-set forests with union by rank and path compression. We start with the forest containing singleton sets $\{i\}$, for $i = 1, \ldots, 20$. We execute the following sequence of operations:

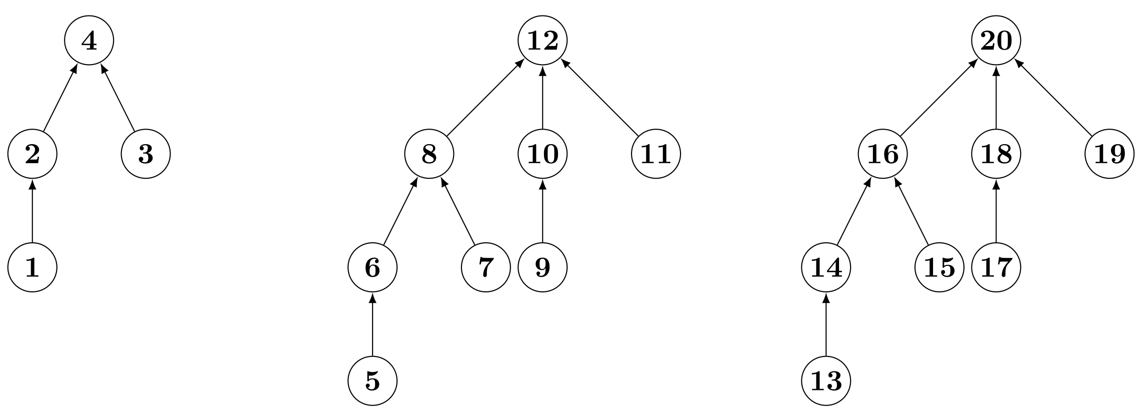
$\mathsf{union}(1,2)$	$\mathsf{union}(3,4)$	$\mathsf{union}(5,6)$	$\mathsf{union}(7,8)$	$\mathsf{union}(9,10)$	$\mathsf{union}(11,12)$	$\mathsf{union}(13,14)$
$\mathsf{union}(15,16)$	$\mathsf{union}(17,18)$	$\mathsf{union}(19,20)$	$\mathsf{union}(2,4)$	$\mathsf{union}(6,8)$	$\mathsf{union}(10,11)$	$\mathsf{union}(13,16)$
$\mathsf{union}(17,20)$	$\mathsf{find}(1)$	$\mathsf{union}(8,12)$	$\mathsf{union}(14,20)$	$\mathsf{union}(2,17)$	$\mathsf{find}(9)$	$\mathsf{union}(13,5)$

Draw the forest (a) after $\mathsf{union}(17,20)$, (b) after $\mathsf{union}(14,20)$, (c) after $\mathsf{union}(13,5)$. Assume that whenever we execute $\mathsf{union}(x,y)$, then we first do $\mathsf{find}(x)$, $\mathsf{find}(y)$, and then the union for the roots. (This slightly differs from the lecture.)

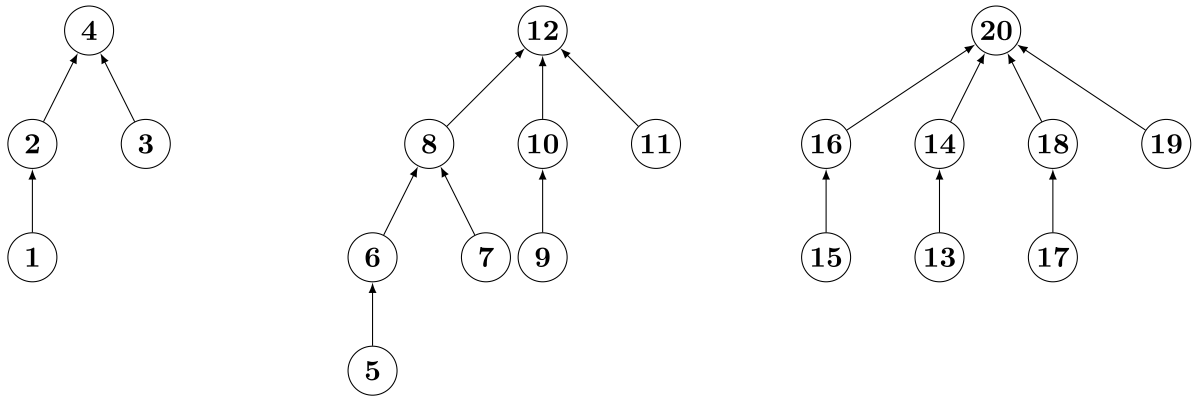
Answer: (a). After $\mathsf{union}(17,20)$, union by rank has a same graph as path compression:



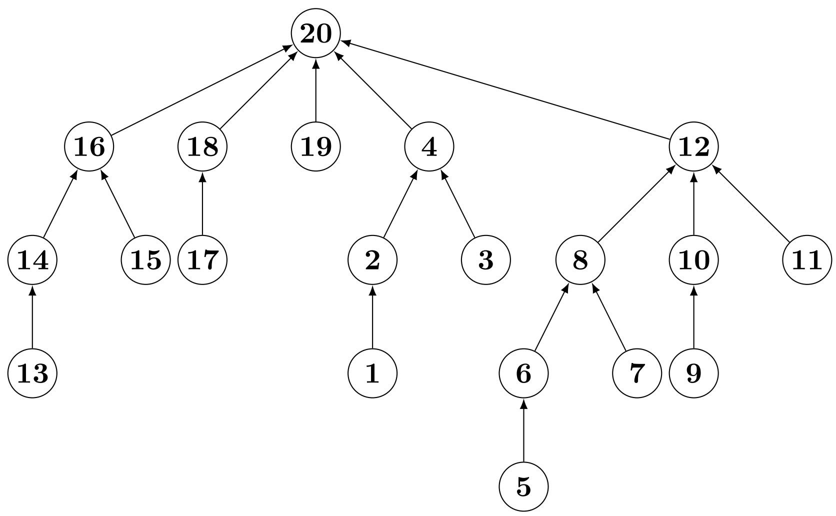
(b). After $\mathsf{union}(14,20)$, union by rank:



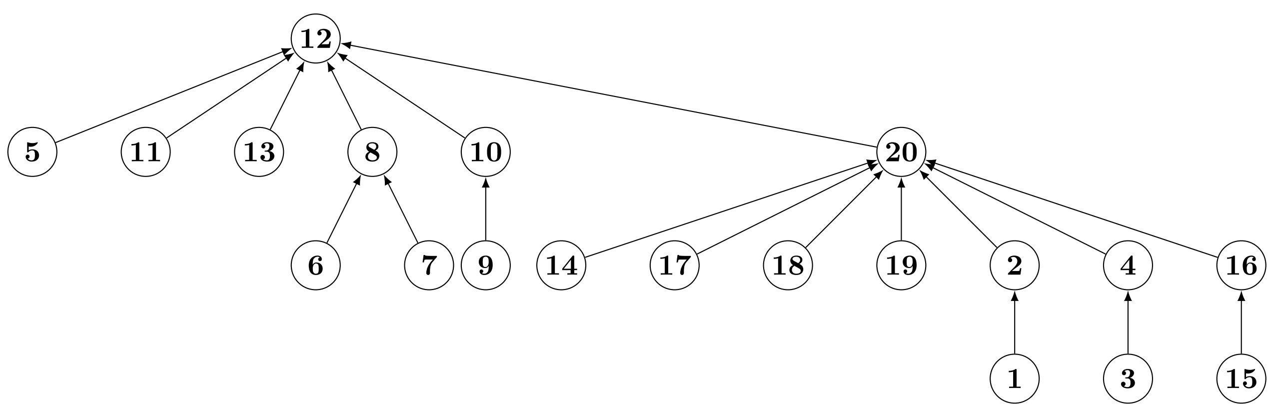
Path compression:



(c). After union$(13, 5)$**, union by rank:**



Path compression:



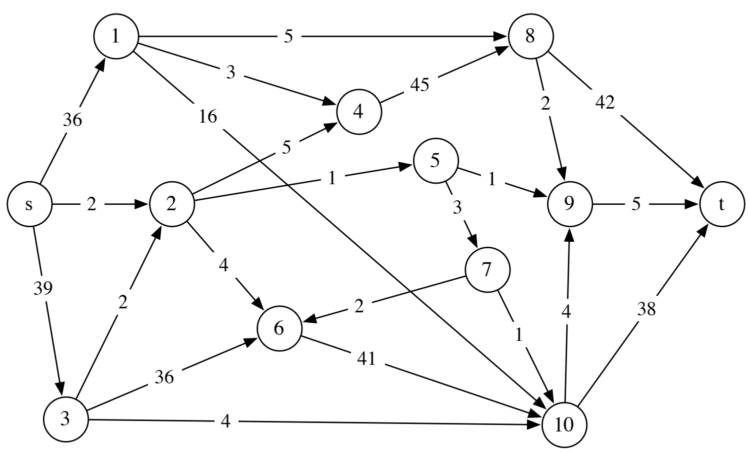
Problem 2: (10 points) Let NetFlow **stand for the network flow problem, as covered in class. Consider a generalized network flow problem** NetFlowVC**, in which capacities are defined both for edges and vertices. In this version, the total flow into a vertex cannot exceed the given capacity. Show that** NetFlowVC **can be reduced to** NetFlow**, and that its**

time complexity is the same.

Answer: The steps are as follow: first we can ocnvert the graph $G = (V, E)$ with node capacities to an equivalent graph G' that only has edge capacities. Second, we need to show that new graph G' satisfies three constraints for the normal max-flow problem.
For each node v with node capacity c_v in G except s, t, we can convert it to an edge (v_{in}, v_{out}). All the edges into v is now connected to v_{in}, and edges out of v is connected to v_{out}. The edge (v_{in}, v_{out}) will have a capacity of c_v.
For capacity constraint, for each node v in G except s, t, $f(v) \leq c_v$. In G', the total incomming flow into node v_{in} is no more than the capacity of edge (v_{in}, v_{out}). For conversation constraint, for each node v in G except $s, , t$, the total flow into v equal to total flow out of it. So, in G', flows into v_{in} equal to flows out of v_{out}, which also equal to the flow from v_{in} to v_{out}. For symmetry constraint, for each (v_{in}, v_{out}) in G', $f(v_{in}, v_{out}) = -f(v_{out}, v_{in})$. Therefore, NetFlowVC can be reduced to NetFlow with same time complexity.

(Reference: `https://cseweb.ucsd.edu/classes/sp16/cse202-a/hw3sol.pdf`)

Problem 3: (10 points) Consider the network below. Use Dinic's algorithm to find the maximum flow in this network. For each phase of the algorithm, except first, you need to show (a) the residual graph before the augmentation, and (b) the layered graph and the flow resulting from the augmentation. At the end, show the final flow and prove that it is indeed maximum.



1 Submission.

Submit the pdf file via gradescope by 5PM on Thursday, February 21. For pair assignments, sub