☐ Exam 2

- Graph Algorithms
  - DFS(G) ⇒ $O(n+m)$
    - outputs vertices of G labeled by connected components
    - Topological sorting by postorder # ⇒ $O(n+m)$
  - Strongly Connected Component
    - v, w strongly connected if path $v \to w$ and $w \to v$
    - SCC(G): ⇒ $O(n+m)$
      1) construct G^R
      2) run DFS on G^R
      3) order V by ↓ post #
      4) run undir connected components alg on G
         - This is undirected G version of DFS
- Minimum Spanning Tree
  - Kruskal's ⇒ $O(|E| \log |V|)$ finds MST
    1) Sort E by ↑ weight
    2) set $X = \emptyset$
    3) for $e = (v, w) \in E$:
       if $X \cup e$ doesn't have cycle: $X = X \cup e$
    4) return X
  - Prim's ⇒ $O(|V| \log |V| + |E| \log |V|)$ finds MST
    1) Pick a node at random
    2) Select next node by smallest edge weight from a visited node to an unvisited
    3) repeat until no nodes remain
  - Cut Property
    - Take $S \subseteq V$ where no edge of X is in cut$(S, \bar{S})$ and $X \subset E$ where $X \subset T$ for a MST T
    - e^* is min weight edge in cut $(S, \bar{S})$
    - $X \cup e^* \subset T'$ where T' is MST
- Dijkstra's ⇒ $O(|E| + |V| \log |V|)$ finds shortest path from one node to every other node
- BFS ⇒ $O(n+m)$ finds dist $y \to z$ where $z \in V$

- Max Flow
  - Send supply $s \to t$
    - maximize total flow
    - don't go over capacity of edges
    - Input: dir $G=(V,E)$ with $s,t \in V$ and caps $c_e > 0$ for $e \in E$
    - Goal: find flows f_e for $e \in E$
      where: for all $e \in E$ $0 \le f_e \le c_e$ (capacity)
      for all $v \in V - \{s \cup t\}$ flow into v = flow out of v
      - want valid flow of max size
- Residual
  - The USED forward capacity is the REMAINING backward capacity for an edge
    - ie. if an edge is using 6/8 capacity, 2 more can be sent forward, 6 can be sent backward

- Ford-Fulkerson $\Rightarrow O(mC)$ where C = size of max flow, $m = |E$
  1) set all $f_e = 0$ for $e \in E$
  2) Build residual network G^f for current flow f
  3) check for st-path p in G^f using BFS or DFS
     3a) if no st-path exists output f
  4) Given p, let $c(p)$ = min capacity along p in G^f
  5) Augment f by $c(p)$ units along p
  6) Repeat from (2) until no st-path
  - assumes integer capacities
  - running time depends on output
  - Pseudo-polynomial

- Edmonds-Karp $\Rightarrow O(m^2 n)$
  - exactly the same alg. as F-F, but MUST use BFS in step 3
- Max-Flow Min-Cut
  - verify f^* is max flow in $O(n+m)$ with DFS on G^f
  - st-cut is cut with $s \in L$ and $t \in R$ for cut(L,R)
    - capacity(L,R) = total flow out of L
    - Max-flow = Min-cut
  - many problems can be reduced to min-cut problem
    - image segmentation example
      - given input (G, f, b, p) for image segmentation
      - define flow network (G', c)
        - add double direction path between each vertical and horizontal pixel
        - add node $s \rightarrow v \in V$
        - add node $t \leftarrow v \in V$
      - get flow f^* of max size
        - $\text{size}(f^*)$ = capacity of min st-cut $= \min_{(F,B)} w'(F,B)$
      - $\max_{(F,B)} w(F,B) = L - \min_{(F,B)} w'(F,B)$
- Max-Flow with demands $\Rightarrow$ find feasible flow
  - input G with caps c and demands d
  - construct G' with $c'(e)$ for max flow
    - for $e \in E$, $c'(e) = c(e) - d(e)$
    - for $v \in V$, add $s' \rightarrow v$ with $c'(\vec{sv}) = d^{in}(v)$
    - for $v \in V$, add $v \rightarrow t'$ with $c'(\vec{vt}) = d^{out}(v)$
    - add $t \rightarrow s$ with $c(\vec{ts}) = \infty$
  - for flow f' in G', $\text{size}(f') \leq D$
    - f' is saturating if $\text{size}(f') = D$
    - D is total capacity out of s', in to t'
  - G has feasible flow iff G' has saturating flow

- exam 2 quick sheet
  - Kruskal's
    - $O(m \log n)$
    - finds MST
  - Prim's
    - $O(m\log n + n\log n) \Rightarrow O(m \log n)$
    - finds MST
  - DFS
    - $O(m+n)$
    - finds connected components
  - BFS
    - $O(m+n)$
    - finds $dist(s,v)$ where $v \in V$
  - Explore
    - $O(m+n)$
    - finds all vertices connected to some vertex z
  - Dijkstra's
    - $O(m + n\log n)$
    - finds $dist(s,v)$ where $v \in V$
  - Ford-Fulkerson
    - $O(mC)$
    - finds max flow of graph (integer capacities)
  - Edmonds-Karp
    - $O(m^2 n)$
    - finds max flow of graph (must use BFS)