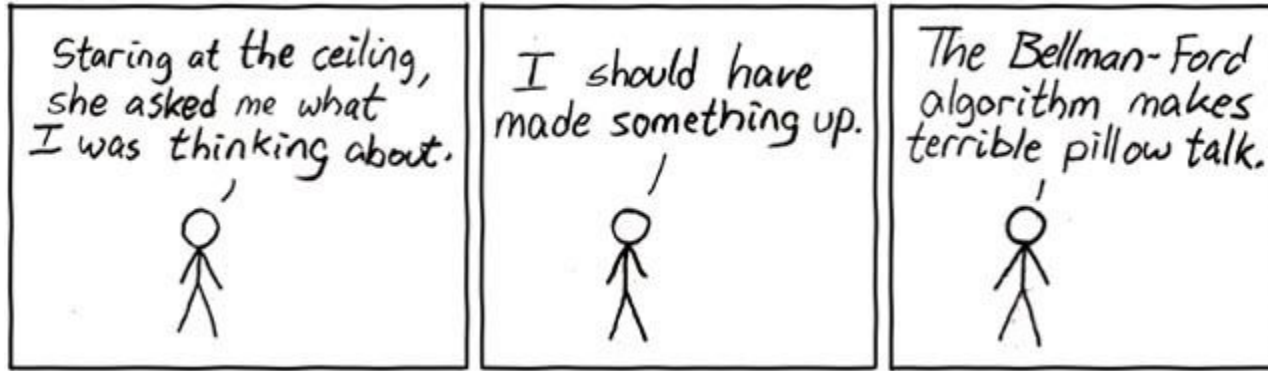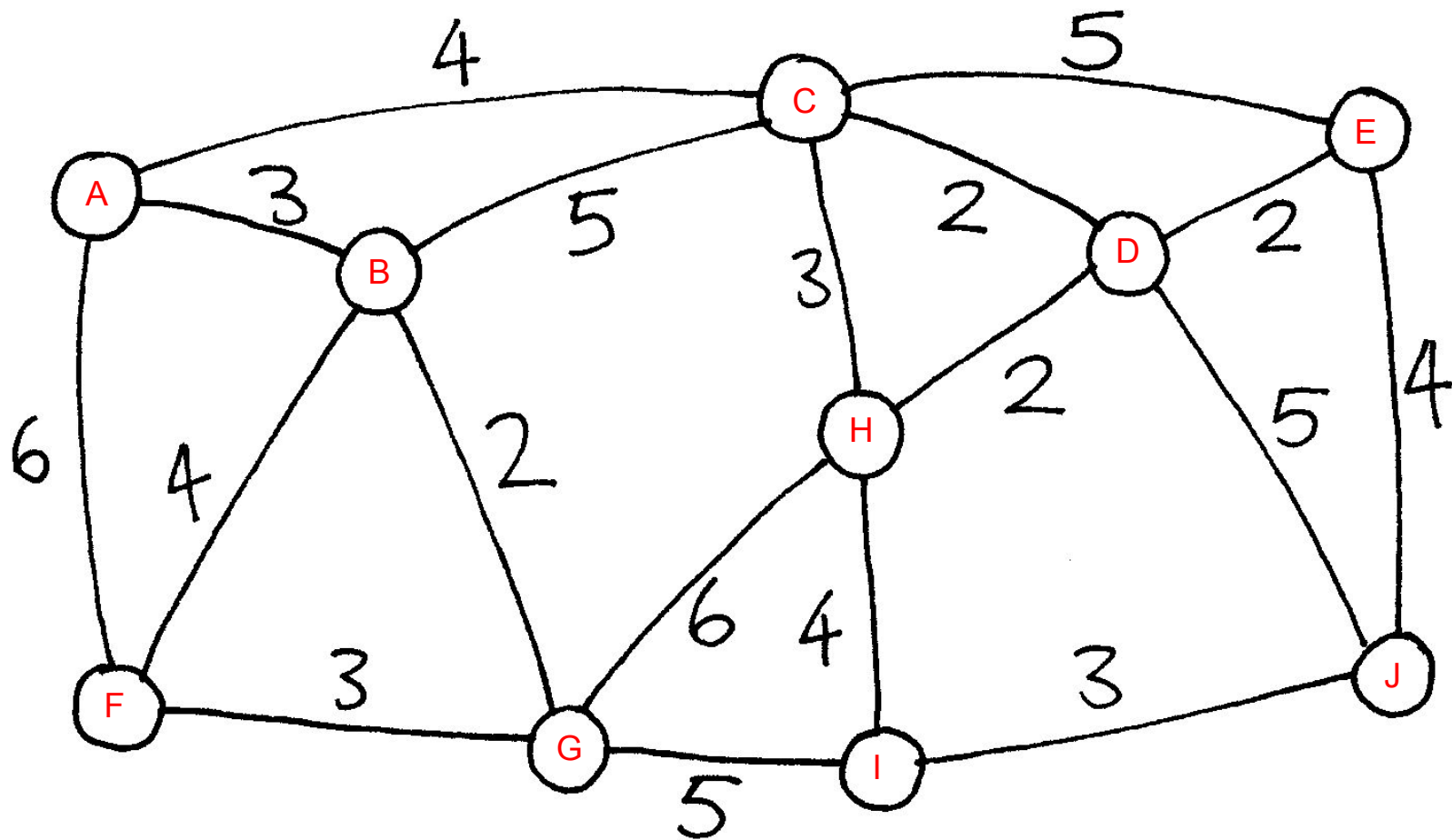# COMP20007 Design of Algorithms: Week 5

- Prepare for MST: be up-to-date on tutorial exercises, and complete the assigned readings
- Week 6 tutorial – revision for MST, incl sample test (if you're in Friday pm tute feel free to attend an additional tute earlier in the week)
- Week 6 workshop – catch up – finish any previous labs

Source: csunplugged.org

Bellman-Ford Algorithm

procedure shortest-paths (G, l, s)

Input: Directed graph $G = (V, E)$;
                     edge lengths $\{l_e : e \in E\}$ with no negative cycles;
                     vertex $s \in V$

Output: For all vertices u reachable from s, dist(u) is set
                     to the distance from s to u.

for all $u \in V$:
    dist$(u) = \infty$
    prev$(u) = $ nil
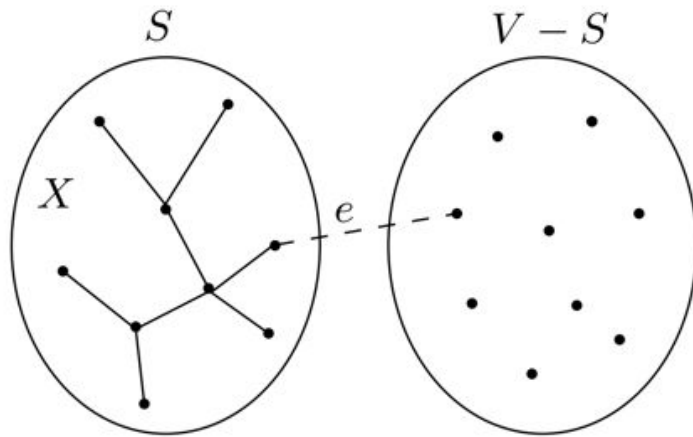
dist$(s) = 0$
repeat $|V| - 1$ times:
    for all $e \in E$:
       update(e)

procedure update$((u, v) \in E)$
dist$(v) = \min\{$dist$(v),$ dist$(u) + l(u, v)\}$

Greedy Algorithms (DPV chapter 5)   *not for MST*

Minimum Spanning Tree revisited: Prim's algorithm

$X = \{ \ \}$ (edges picked so far)
repeat until $|X| = |V| - 1$:
    pick a set $S \subset V$ for which X has no edges between S and $V - S$
    let $e \in E$ be the minimum-weight edge between S and $V - S$
    $X = X \cup \{e\}$

Minimum Spanning Tree revisited: Prim's algorithm

```
procedure prim(G, w)
Input:    A connected undirected graph G = (V, E) with edge weights we
Output:   A minimum spanning tree defined by the array prev

for all u ∈ V:
    cost(u) = ∞
    prev(u) = nil
Pick any initial node u0
cost(u0) = 0

H = makequeue(V)      (priority queue, using cost-values as keys)
while H is not empty:
    v = deletemin(H)
    for each {v, z} ∈ E  && z in H
        if cost(z) > w(v, z):
            cost(z) = w(v, z)
            prev(z) = v
            decreasekey(H, z)
```

WARNING! Fig 5.9 in book missing key detail.

What was the property we checked last time, and why aren't we checking it this time? Why is this better?

Dijkstra, Bellman-Ford, Prim...

A	B	C	D	E	F	G	H	I	J
∞, nil	∞, nil	∞, nil	∞, nil	∞, nil	∞, nil	∞, nil	∞, nil	∞, nil	∞, nil