Worksheet 9

1. The given function can be rewritten as

$$\begin{aligned} sub(x,0) &= x \\ sub(x,y+1) &= p(sub(x,y)) \end{aligned}$$

Proof. by induction on y.

Base case $y = 0$

$$\begin{aligned} sub(x,0) &= x - 0 \\ &= x \end{aligned}$$

Induction step $y \to y+1$

$$\begin{aligned} sub(x,y+1) &= x - y - 1 \\ &= sub(x,y) - 1 \quad \text{(induction hypothesis)} \\ &= p(sub(x,y)) \end{aligned}$$

□

2. With the definition

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

we can plug in the numbers to see that

$$\begin{aligned} A_2(j) &= A_1^{(j+1)}(j) \\ &= 2^{j+1}(j+1) - 1. \end{aligned}$$

Proof. $A_1^{(n)}(j) = 2^n(j+1) - 1$

Base case $n = 1$

$$A_1(j) = 2j + 1 = 2^1(j+1) - 1$$

Induction step $n \to n+1$

$$\begin{aligned} A_1^{(n+1)}(j) &= A_1(A_1^{(n)}(j)) \\ &= 2A_1^{(n)}(j) + 1 \\ &= 2(2^n(j+1) - 1) + 1 \\ &= 2^{n+1}(j+1) - 2 + 1 \\ &= 2^{n+1}(j+1) - 1 \end{aligned}$$

□

3.

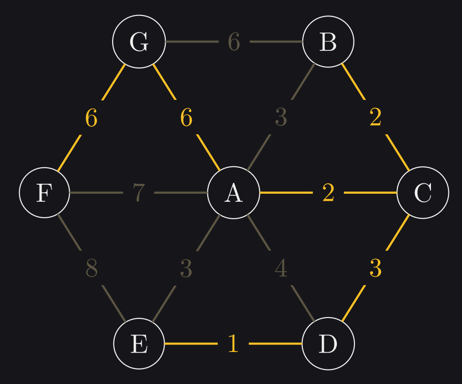

Figure 1: spanning tree with cost of 20

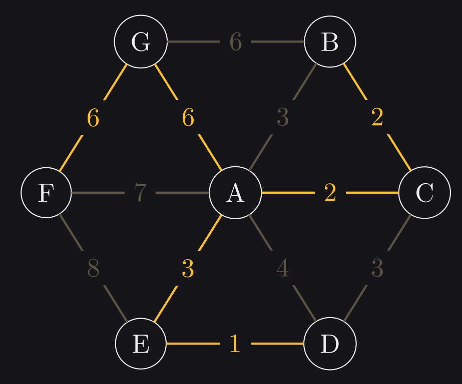

Figure 2: spanning tree with cost of 20

For this labling, the steps of the Kruskal's algorithm would look like this: Pick any edge that has the minimal cost with at most one node in the current set.

#1 Pick (E, D).

#2 Pick (B, C).

#3 Pick (A, C).

#4 Pick (C, D).

#5 Pick (A, G).

#6 Pick (G, F).

This spanning (shown in the figure 3) tree is not unique, we can see that there's at least one other spanning tree with the same cost in figure 2.

4. (a) Let V be the set of all vertices and E the edges of an undirected graph. Let S_i be the i-th connected component of (V, E) and let $S = \bigcup_{i=1}^{n} S_i$. Start by picking a vertex v_1 from $V \setminus S$ and put all the vertices visited by $\text{dfs}(v_1)$ into S_1. Then, pick next vertex v_2 from $V \setminus S$ and put all the vertices visited by $\text{dfs}(v_2)$ into S_2 and so on.

(b) We need to modify dfs a little for this task:

$$\text{dfs}(v, V, D, E) = \left(\bigcirc_{u:\{u,v\} \in E, u \in V} \left((u) \circ \text{dfs}(u, V \setminus \{u\}, D \cup \{u\}, E) \circ (u) \right) \right) \circ \left(\bigcirc_{u:\{u,v\} \in E, u \in D} (u, v) \right).$$

This will traverse each edge once in each direction.

5. (a) Right here:

$$\begin{aligned} &\vdots \\ d^t[y] &= \delta[y] && \text{(above)} \\ &\leq \delta[y] && \text{(edge weights non-negative)} \\ &< d^t[u] && \text{(assumption)} \\ &\vdots \end{aligned}$$

(b)

Figure 3: An example of a graph with positive and negative edge weights, where Dijkstras algorithm gives the wrong answer