Eric Bronner, Aedan Dispenza, Timothy Yong, Jason Davis
CS-513 - Dr. Farach-Colton

Homework 1

Problem 1

$d + 1 \leq n \leq 2^{d+1} - 1$

Proof by Induction:

1. Base Case:
   tree of $n = 1$, $d = 0$.

$$0 + 1 \leq 1 \leq 2^{0+1} - 1$$
$$1 \leq 1 \leq 1$$

2. Inductive Hypothesis:
   Assume the number of nodes n in a completely sparse tree $n(d-1) = d$.

3. Inductive Step:

$$n(d) = n(d-1) + 1$$
$$n(d) = d + 1 \;\square$$

Proof by Induction:

1. Base Case:
   tree of $n = 1$, $d = 0$.

$$\log n \leq d \leq n - 1$$
$$log 1 \leq 0 \leq 1 - 1$$
$$0 \leq 0 \leq 0$$

2. Inductive Hypothesis:
   Assume we have two subtrees of the same depth
   and the number of nodes $n(d-1) = 2^{(d-1)+1} - 1$ for each subtree,
   thus $n(d-1) = 2^d - 1$.

3. Inductive Step:

$$n(d) = n(d-1) \times 2 + 1$$
$$n(d) = (2^d - 1) \times 2 + 1$$
$$n(d) = 2 \times 2^d - 2 + 1$$
$$n(d) = 2^{d+1} - 1 \;\square$$

where the $\times 2$ is derived from the two subtrees,
and the $+1$ from the root node.

Problem 2

a) $T(1) = 1$, $T(n) = T(\frac{n}{2}) + 1$ for $n = 2^k$, $k \in \mathbb{N}$

$T(n) = T(\frac{n}{2}) + 1$
$= (T(\frac{n}{2^2}) + 1) + 1$
$= ((T(\frac{n}{2^3}) + 1) + 1) + 1$
repeat this recursion k times...
$= T(\frac{n}{2^k}) + k$
$= T(\frac{2^k}{2^k}) + k$ as $n = 2^k$
$= T(1) + k$
$= 1 + k$
$= 1 + logn$

b) $T(1) = 1$, $T(n) = 2T(\frac{n}{2}) + 1$ for $n = 2^k$, $k \in \mathbb{N}$

$T(n) = 2T(\frac{n}{2}) + 1$
$= 2(2T(\frac{n}{2^2}) + 1) + 1$
$= 2^2T(\frac{n}{2^2}) + 2 + 1$
$= 2(2^2T(\frac{n}{2^3}) + 2 + 1) + 1$
$= 2^3T(\frac{n}{2^3}) + 4 + 2 + 1$
repeat this recursion k times...
$= 2^kT(\frac{n}{2^k}) + 2^k - 1$
$= 2^kT(1) + 2^k - 1$
$= 2 \times 2^k - 1$
$= 2n - 1$

c) $T(1) = 1$, $T(n) = 2T(\frac{n}{2}) + n$ for $n = 2^k$, $k \in \mathbb{N}$

$T(n) = 2T(\frac{n}{2}) + n$
$= 2(2T(\frac{n}{2^2}) + \frac{n}{2}) + n$
$= 2^2T(\frac{n}{2^2}) + n + n$
$= 2(2^2T(\frac{n}{2^3}) + \frac{n}{2} + \frac{n}{2}) + n$
$= 2^3T(\frac{n}{2^3}) + n + n + n$
$= 2^kT(\frac{n}{2^k}) + kn$
$= 2^kT(1) + kn$
$= 2^k + kn$
$= n + nlogn$

Problem 3

a) $\sum_{i=1}^{k} i(i+1) = k(k+1)(k+2)/3$

Base Case P(1):

$$1(1+1) = \frac{1(1+1)(1+2)}{3}$$
$$2 = \frac{6}{3}$$
$$2 = 2$$

Inductive Step:

$$T(k+1) = T(k) + (k+1)((k+1)+1) = T(k) + (k+1)(k+2)$$
$$\frac{(k+1)((k+1)+1)((k+1)+2)}{3} = \frac{k(k+1)(k+2)}{3} + (k+1)(k+2)$$
$$\frac{(k+1)(k+2)(k+3)}{3} = \frac{k(k+1)(k+2)}{3} + (k+1)(k+2)$$
$$\frac{k(k+1)(k+2)}{3} + \frac{3(k+1)(k+2)}{3} = \frac{k(k+1)(k+2)}{3} + (k+1)(k+2)$$
$$\frac{k(k+1)(k+2)}{3} + (k+1)(k+2) = \frac{k(k+1)(k+2)}{3} + (k+1)(k+2) \ \square$$

b) $\sum_{i=0}^{k} i2^i = (k-1)2^{k+1} + 2$

Base Case P(0):

$$0(2^0) = (0-1)2^{0+1} + 2$$
$$0 = -2^1 + 2$$
$$0 = 0$$

Inductive Step:

$$T(k+1) = T(k) + (k+1)2^{k+1}$$
$$(k)2^{(k+1)+1} + 2 = (k-1)2^{k+1} + 2 + (k+1)2^{k+1}$$
$$k2^{k+2} + 2 = k2^{k+1} - 2^{k+1} + 2 + k2^{k+1} + 2^{k+1}$$
$$4k2^k + 2 = 2k2^{k+1} + 2$$
$$4k2^k + 2 = 4k2^k + 2 \ \square$$

c)$\sum_{i=0}^{k} \frac{i}{2^i} = 2 - \frac{(k+2)}{2^k}$

Base Case P(0):

$$\frac{0}{2^0} = 2 - \frac{(0+2)}{2^0}$$
$$0 = 2 - \frac{2}{1}$$
$$0 = 0$$

Inductive Step:

$$T(k+1) = T(k) + \frac{(k+1)}{2^{k+1}}$$
$$2 - \frac{((k+1)+2)}{2^{k+1}} = 2 - \frac{(k+2)}{2^k} + \frac{(k+1)}{2^{k+1}}$$
$$\frac{-k-1-2}{2^{k+1}} = \frac{-k-2}{2^k} + \frac{k+1}{2^{k+1}}$$
$$\frac{-k-1}{2^{k+1}} + \frac{-2}{2^{k+1}} = \frac{-k-2}{2^k} + \frac{k+1}{2^{k+1}}$$
$$-2\frac{k+1}{2^{k+1}} + \frac{-2}{2^{k+1}} = \frac{-k-2}{2^k}$$
$$\frac{-k-1}{2^k} + \frac{-1}{2^k} = \frac{-k-2}{2^k}$$
$$\frac{-k-2}{2^k} = \frac{-k-2}{2^k} \; \square$$

Problem 4
Place the following in increasing asymptotic order:
$4n, n^2, n \log n, n \ln n, \log n, e^n$

First we, drop all constants, so $4n = O(n)$.
$O(\log n) < O(n) < O(n^2) < O(e^n)$.
$O(n \log n)$ and $O(n \ln n)$ are both greater than $O(n)$ and less than $O(n^2)$.

$$n \log n > n \ln n$$
$$\log n > \ln n$$

Therefore, the correct order is:
$\log n, 4n, n \ln n, n \log n, n^2, e^n$.

Problem 5
Proof by Contradiction:
Say we have T and $T*$, both are $MSTofG$ and $T \neq T*$
Because $\forall e,\ e' \in E,\ w(e) \neq w(e')$ where w(e) is the weight of e,
all edges can be identified by their unique weight
Say $e \in E \in T,\ e' \in E \in T*$, where $e,\ e' \in E \in G$

Case 1: $w(e) > w(e')$
then $w(T) > w(T*)$, where w(T) is the sum of all weights
therefore, T is not an MST of G, which is a contradiction

Case 2: $w(e) < w(e')$
then $w(T) < w(T*)$, where w(T) is the sum of all weights
therefore, T* is not an MST of G, which is a contradiction

Case 3: $w(e) = w(e')$
then $w(T) = w(T*)$, where w(T) is the sum of all weights
therefore, T = T*, which is a contradiction $\square$

Problem 6
Given a matrix-representation of a graph, and the following functions:
Connect(u,v) which runs in $O(1)$
Disconnect(u,v) which runs in $O(1)$

Adj(u,v) which runs in $O(1)$
we have developed the following algorithm:

```
HamC(G):
    Adj(u,v):
        Disconnect(u,v)
        HamC(G):
            FALSE
        Else:
            TRUE
    Else:
        FALSE
Else:
    Connect(u,v)
    HamC(G):
        TRUE
    Else:
        FALSE
```

Due to the use of HamC(G), this algorithm runs in $O(n)$.