

**CMPS 101**  
**Fall 2010**  
**Homework Assignment 7**

1. (1 Point) p.1091: B.5-4

Use induction to show that a nonempty binary tree with  $n$  nodes and height  $h$  satisfies  $h \geq \lfloor \lg n \rfloor$ .

Hint: use the recursive definition of height discussed in class:

$$h(T) = \begin{cases} -\infty & n(T) = 0 \\ 0 & n(T) = 1 \\ 1 + \max(h(L), h(R)) & n(T) > 1 \end{cases}$$

Here  $n(T)$  denotes the number of nodes in a binary tree  $T$ ,  $h(T)$  denotes its height,  $L$  denotes its left subtree, and  $R$  its right subtree. Note that this proof can be phrased equally well as an induction on  $n(T)$  or on  $h(T)$ . Additional hint: use (and prove) the following fact:  $\lfloor \lg(2k+1) \rfloor = \lfloor \lg(2k) \rfloor$  for any positive integer  $k$ .

2. (1 Point)

Let  $T$  be an almost complete binary tree on  $n$  nodes, and let  $N(n, h)$  denote the number of nodes in  $T$

at height  $h$ , where  $0 \leq h \leq \lfloor \lg(n) \rfloor$ . Prove that  $N(n, h) = \left\lfloor \frac{n}{2^h} \right\rfloor - \left\lfloor \frac{n}{2^{h+1}} \right\rfloor$ . Hint: first observe that the

number of leaves in  $T$  is  $n - \left\lfloor \frac{n}{2} \right\rfloor$ .

3. (1 Point) p.132: 6.2-5

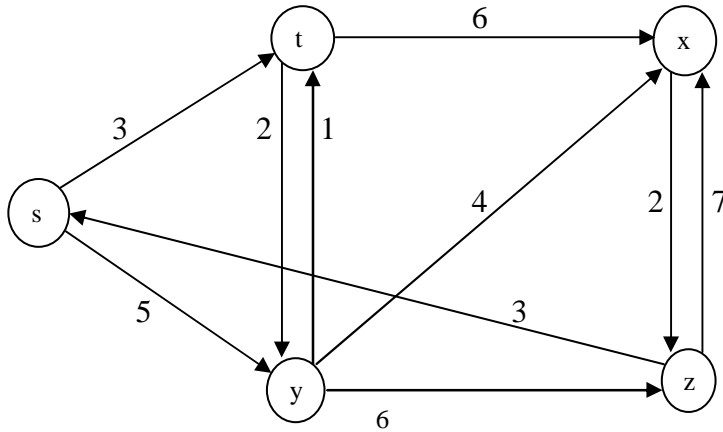
The code for Max-Heapify is quite efficient in terms of constant factors, except possibly for the recursive call in line 10, which might cause some compilers to produce inefficient code. Write an efficient Max-Heapify that uses an iterative control construct (a loop) instead of recursion.

4. (1 Point)

Let  $x \in V(G)$  and suppose that after INITIALIZE-SINGLE-SOURCE( $G, s$ ) is executed, some sequence of calls to Relax( ) causes  $d[x]$  to be set to a finite value. Then  $G$  contains an  $s$ - $x$  path of weight  $d[x]$ . (Hint: Use induction on the length of the Relaxation sequence.)

5. (1 Point) p.600: 24.3-1

Run Dijkstra's algorithm on the directed graph of Figure 24.2 (pictured below), first using vertex  $s$  as the source and then using vertex  $z$  as the source. Show the  $d$  and  $\pi$  values and the vertices in set  $S$  after each iteration of the **while** loop.



6. (1 Points) p.600: 24.3-4

We are given a directed graph  $G=(V,E)$  on which each edge  $(u,v) \in E$  has an associated value  $r(u,v)$ , which is a real number in the range  $0 \leq r(u,v) \leq 1$  that represents the reliability of a communication channel from vertex  $u$  to vertex  $v$ . We interpret  $r(u,v)$  as the probability that the channel from  $u$  to  $v$  will not fail, and we assume that these probabilities are independent. Give an efficient algorithm to find the most reliable path between two given vertices.