Computer Science 477/677 Spring 2019

University of Nevada, Las Vegas Computer Science 477/677 Spring 2019

Practice for Third Examination April 17, 2019

1. Find the asymptotic complexity, in terms of n, for each of these fragments, expressing the answers using O, Θ, or Ω, whichever is most appropriate.

   (a)
   ```
   for(int i = 1; i*i < n; i++)
     cout << "Hi!" << endl;
   ```

   (b)
   ```
   for(int i = n; i > 1; i = sqrt(i));
     cout << "Hi!" << endl;
   ```

2. Find the asymptotic time complexity, in terms of n, for each of these code fragments, expressing the answers using O, Θ, or Ω, whichever is most appropriate.

   (a)
   ```
   int f(int n)
     {
      if (n < 2) return 1;
      else return f(n-1)+f(n-1);
     }
   ```

   (b)
   ```
   void hello(int n)
     {
      if(n >= 1)
       {
        for(int i = 1; i < n; i++)
         cout << "Hello!" << endl;
        hello(n/2);
        hello(n/2);
       }
     }
   ```

3. The sequence of *Kresge numbers* $K_1, K_2, K_3, \ldots$ is defined recursively by:

   (a) $K_1 = 1$

   (b) If $n > 1$, then $K_n = K_{n-1} + K_{n/2}$, where $n/2$ is an integer obtained by truncating as in C++: for example $5/2 = 2$ and $6/2 = 3$.

   Write, in pseudocode, a non-recursive algorithm which computes K_{100}.

4. The sequence of *von Drachenfels numbers* $V_1, V_2, V_3, \ldots$ is defined recursively by:

   (a) $V_1 = 1$

   (b) $V_2 = 1$

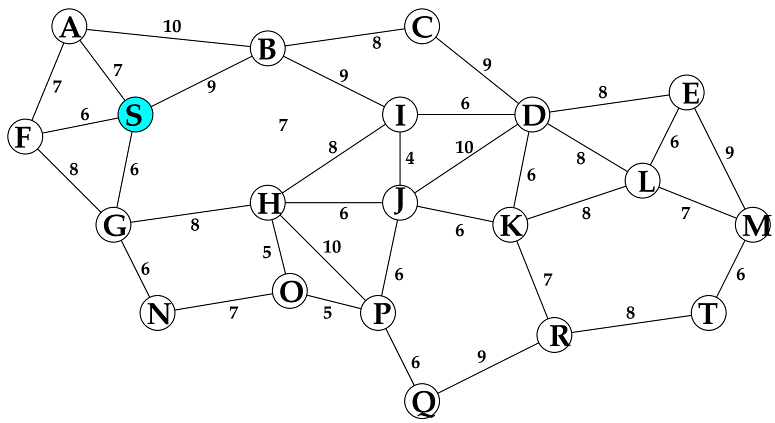
   (c) If $n > 2$, then $V_n = V_{n/3} + V_{n/2}$, where $n/3$ and $n/2$ are integers obtained by truncating as in C++.

   Describe a memoization algorithm which computes V_{100000}.

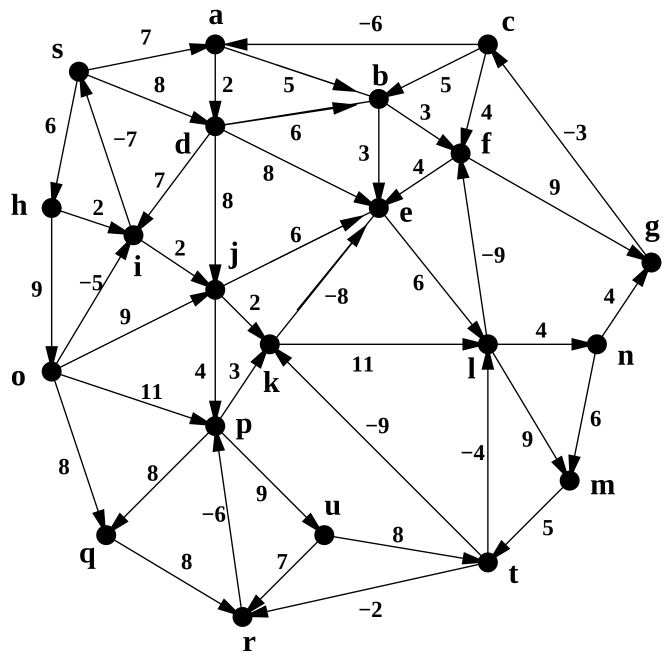
5. Give pseudocode for a recursive algorithm which computes the median of the union of two sorted lists in logarithmic time.

6. Describe a randomized algorithm which finds the k^{th} smallest element of an unsorted list of n distinct numbers, for a given $k \le n$, in $O(n)$ expected time. (By "distinct," I mean that no two numbers in the list are equal.)

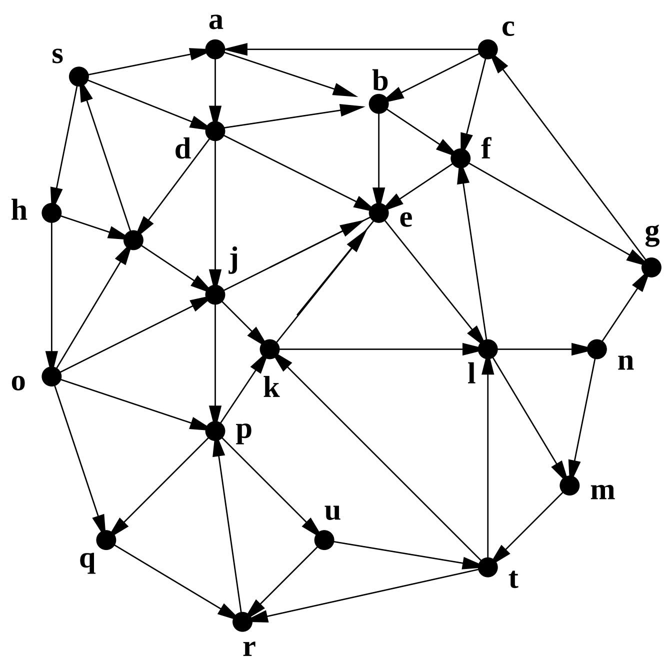
7. Walk through Dijkstra's algorithm for the following weighted graph to solve the single source shortest pair problem, where S is the source.



8. The first step of Johnson's algorithm is to compute the heuristic function On the weighted directed graph (a) below, label each node of (a) with the correct heuristic. (You do not have to show the steps of the algorithm for this. The example is small enough that you can simply compute the values in your head.) The next step is to adjust the arc weights. Label the arcs of (b) with the adjusted weights.

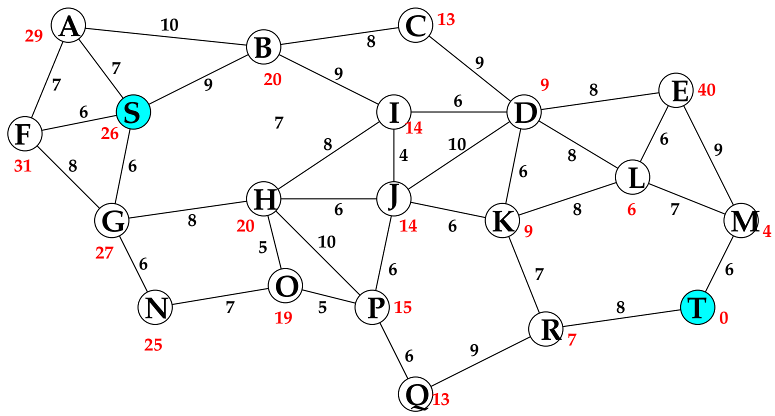


(a)



(b)

9. Walk through the A^* algorithm for the following weighted graph to find the shortest path from S to T. Edge weights are shown in black, and the values of the heuristic are shown in red.



Use these figures for working out Problems 7, 9. Make multiple copies of this page if needed.