Algorithm
Comprehensive Exam
Fall 2017

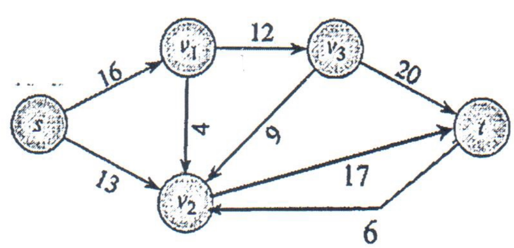
Short Questions (answer any six. Each carries 7 points)

1. Derive the tight lower and upper bounds of the following recurrence:
   $T(n) = 3 \cdot T(n/4) + n^{1/2}$

2. The utilization efficiency of a hash table depends heavily on its hashing function(s) employed. Describe with a diagram to illustrate how a multiplication method of hashing works on a machine with the word size of w bits for a hash table with 2^p entries, $p < w$.

3. Show your construction of an optimal Huffman code for the set of 7 frequencies: a:2 b:4 c:5 d:11 e:23 f:36 g:54.

4. The Edmonds-Karp algorithm (*EK*) follows the basic Ford-Fulkerson method with breadth-first search to choose the shortest augmenting path (in terms of the number of edges involved) for computing the maximum flow iteratively from vertex s to vertex t in a weighted directed graph, where the capacity of each link is given. Illustrate the maximum flow computation process (including the augmenting path chosen in each iteration and its resulting residual network) via *EK* for the graph depicted below.



5. How should we construct a binary search tree that minimizes the average search (access) time of n distinct keys where $n = 2^{k+1} - 1$ (for some integer $k >= 0$) where each key is accessed with the same probability? Provide a pseudo-code to create such tree for a set of available keys.

6. What are the three basic steps in divide and conquer strategy? If quicksort is a divide and conquer algorithm, instantiate the general steps of divide and conquer algorithm to quick sort. Otherwise, rationalize why quick sort is not divide and conquer algorithm.

7. Mark True or False against all the following statements:
   a. A binary search tree of size N will always find a key at most O(log N) time
   b. An optimal binary search is not necessarily a balanced tree
   c. A binary heap always maintains a balanced tree as practical as it can be.
   d. To implement a priority queue binomial heap is preferred over binary heap.
   e. A graph formed by strongly connected component nodes, a strongly connected component graph (SCC), is always a minimum spanning tree.

Long Question (Answer any three. Each carries 20 points)

1. Consider the matrix-chain multiplication problem for four matrices A_1, A_2, A_3, A_4, with their sizes being 30×10, 10×20, 20×50, and 50×40, respectively. Follow the tabular, bottom-up method in the procedure of MATRIX-CHAIN-ORDER below to construct tables that keep respectively entry $m[i, j]$ for all $1 \le i, j \le 4$ and entry $s[i, j]$ for $1 \le i \le 3$ and $2 \le j \le 4$ to get the optimal parenthesized multiplication result.
   (a) Construct the two tables, with their entry values shown.
   (b) Show the parenthesized multiplication of the matrix-chain.

```
MATRIX-CHAIN-ORDER(p)
n = p.length − 1
let m[1..n, 1..n] and s[1..n − 1, 2..n] be new tables
for i = 1 to n
    m[i, i] = 0
for l = 2 to n                // l is the chain length
    for i = 1 to n − l + 1
        j = i + l − 1
        m[i, j] = ∞
        for k = i to j − 1
            q = m[i, k] + m[k + 1, j] + p_{i−1} p_k p_j
            if q < m[i, j]
                m[i, j] = q
                s[i, j] = k
return m and s
```

2. A. Explain what do you understand by "principle of optimality"

   B. Write down the basic rule that satisfy the principle of optimality and domain related constraints to the following problems.
      B1. 0-1 knapsack problem.
      B2. Pairwise shortest path problem.
      B3. Chain matrix multiplication problem.

   C. 0-1 knapsack problem is a member of NP-complete. John, a fellow student, argues that dynamic programming solves 0-1 knapsack problem optimally, all the NP-complete problems have polynomial solutions. Comment on John's statement related to polynomial solution to NP-complete problem.

3 a. Briefly describe NP-class, P-class, NP-complete and NP-hard.

  b. Show the conjectured relationship among the classes NP-class, P-class, NP-complete and NP-hard.

  c. Show that sorting n objects with integer key values belongs to NP-class.

  d. Provide the steps involved in showing whether a problem belongs to NP-complete or not.
     (i) Illustrate the steps in step d by showing 3 proposition satisfiability (3-p sat) problem belongs to NP-complete.

     (ii) Provide a pseudo code that attempt to solve 3-p sat problem heuristically.

4.
A. In the following pseudo code, write the formula to determine number of add operations at line 5 when the algorithm terminates. What is the space complexity? Find the following time bounds of this algorithm: upper, lower and tight (must show all the details of your work)

```
Procedure Count()
1 Cnt=0
2 For i = 1 to n do {
3       For j= 1 to i do {
4         For k= 1 to j do {
5           Cnt = Cnt + 1 }}}
```

B. Suppose a divide and conquer algorithm, say Alg, accepts a string of length n as input and creates two strings of length (n-1) in some constant time C_1 and it call itself (Alg) for each of these strings. The algorithm Alg terminates when the length of the string is 1. The time taken by the algorithm (Alg) to combine the results is a constant time, say C3.
  I. Construct a recurrence relation for its runtime T(n)
  II. Solve for T(n)