國立台灣海洋大學資訊工程學系

103 學年度第 1 學期資格考　科目：演算法 (1/2)

1. (20%) Explain the following terms:
   (a) Complete bipartite graph
   (b) Euler cycle
   (c) Optimal substructure property
   (d) NP-complete

2. (10%) Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = 3T(\lfloor n/4 \rfloor) + n^2$. Use the substitution method to verify your answer.

3. (15%) Given a sequence $X = \langle x_1, x_2, \ldots, x_n \rangle$, the longest increasing subsequence problem is to find an increasing subsequence of X with the longest length.
   (a) Give an algorithm that solves the longest increasing subsequence problem in $O(n^2)$ time.
   (b) Analyze the time complexity of your algorithm.
   (c) Use your algorithm to find a longest increasing subsequence for
   $X = \langle 5, 1, 8, 12, 9, 11, 18, 35, 17 \rangle$.

4. (15%) Please answer the following questions.
   (a) What is the running time of **quicksort** when all elements have the same value? Justify your answer.
   (b) Show that **quicksort**'s average-case running time is $O(n \lg n)$.
   (c) Show that **quicksort**'s worse-case running time is $O(n^2)$ and how **quicksort** can be made to run in $O(n \lg n)$ time in the worst case.

5. (10%) Let S and T be two n-element arrays sorted in nondecreasing order.
   (a) Give an algorithm that finds the median of all $2n$ numbers whose time complexity is $O(\lg n)$.
   (b) Analyze the time complexity of your algorithm.

6. (10%) Please answer the following questions.
   (a) Define knapsack problem.
   (b) Solve the continuous-knapsack problem for the given weights w_i's, profits p_i's, and knapsack capacity =140. Tell which objects are chosen and give the maximum profit.

i	1	2	3	4	5	6
w_i	20	50	60	15	20	30
p_i	2	3	4	4	1	6

7. (20%) Please answer the following questions.
   (a) What are the properties of heap?
   (b) Show that the height of an n-node binary heap is $\lfloor \lg n \rfloor$.
   (c) Analyze the time complexity of the heapify algorithm below.
   (d) Trace the operation of heapify on the following input array
   21, 94, 88, 64, 17, 54, 38, 22, 7, 26

```
heapify(v, n) {
  // n/2 is the index of the parent of the last node
  for i = n/2 downto 1
    siftdown(v, i, n)
}

siftdown(v, i, n) {
  temp = v[i]
  // 2 * i ≤ n tests for a left child
  while (2 * i ≤ n) {
    child = 2 * i
    // if there is a right child and it is bigger than the left child, move child
    if (child < n && v[child + 1] > v[child])
      child = child + 1
    // move child up?
    if (v[child] > temp)
      v[i] = v[child]
    else
      break // exit while loop
    i = child
  }
  // insert original v[i] in correct spot
  v[i] = temp
}
```