Xi'an Jiaotong-Liverpool University

Xi'an Jiaotong-Liverpool University
西交利物浦大學

| PAPER CODE | EXAMINER | DEPARTMENT | TEL |
|---|---|---|---|
| CSE 204 | | Computer Science and Software Engineering | |

**2017/18 SEMESTER 2 – FINAL EXAMINATIONS**

**BACHELOR DEGREE – Year 3**

**Complexity of Algorithms**

**TIME ALLOWED: 2 Hours**

---

**INSTRUCTIONS TO CANDIDATES**

1. **Total marks available are 100. This accounts for 80% of the final mark.**
2. **The number in the column on the right indicates the marks for each question.**
3. **Answer all questions.**
4. **Answers should be written in English in the answer script provided.**
5. **Relevant and clear steps should be included in the answers.**

**Notes:**

- **To obtain full marks for each question, relevant and clear steps should be included in the answers.**
- **Partial marks may be awarded depending on the degree of completeness and clarity.**

**Question 1: Algorithm Analysis [20 marks]**

1. Consider the following code fragment.

```
for i=1 to n do
   for j=i to 2*i do
       output foobar
```

Let T(n) denote the number of times `foobar' is printed as a function of n.

1) Express T(n) as a summation (actually two nested summations). [3 marks]

2) Simplify the summation and give the worst-case running time using Big Oh notation. [6 marks]

2. The processing time of a sorting algorithm is described by the following recurrence equation (c is a positive constant):

$T(n) = 3T(n/3) + 2cn;$

$T(1) = 0$

1) Solve this equation to derive an explicit formula for T(n) assuming $n = 3m$ and specify the Big-Oh complexity of the algorithm. [5 marks]

2) Find out whether the above algorithm is faster or slower than usual Mergesort that splits recursively an array into only two subarrays and then merges the sorted subarrays in linear time cn. [6 marks]

**Question 2: Binary Search Tree and Heap [15 marks]**

1. Specify the worst-case Big-Oh complexity of the following algorithms, assuming an input array of size N: HeapSort, MergeSort, QuickSort, BinarySearch. [2 marks]

2. 1) Convert an array [10, 26, 52, 76, 13, 8, 3, 33, 60, 42] into a maximum heap H. Draw the initial array and the percolation steps of creating the maximum heap. [3 marks]

2) Explain how to convert the above array into a well-balanced binary search tree. Draw the converted well-balanced binary search tree. [4 marks]

3) Explain what is the basic difference between a heap and a binary search tree? [3 marks]

4) Give an algorithm that checks if a given binary tree is a binary search tree. Prove the termination and correctness of your algorithm and discuss its complexity. [3 marks]

**Question 3: Graph [15 marks]**

1. For the graph shown below, answer the following questions:

1) Is the graph connected, strongly connected, or unconnected? [2 marks]

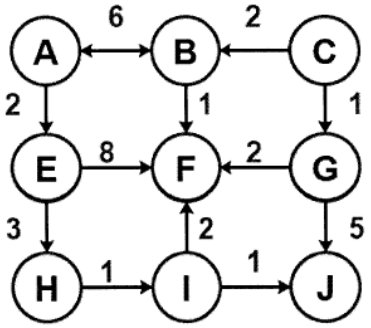
If it is not connected, give an example to explain why not..

2) Is the graph cyclic or acyclic? [1 marks]

If it is cyclic, give an example of a cycle.

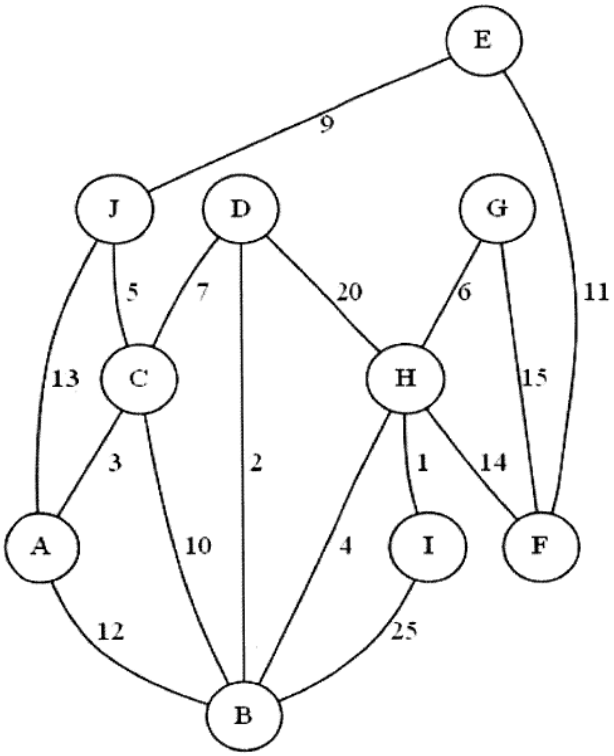
3) Write a complete adjacency list and adjacency matrix representation of the graph. [3 marks]

4) Write the shortest path that a breadth-first search (BFS) would find from vertex A to vertex I. Assume that any foreach loop over neighbors returns them in ABC order. [2 marks]




2. Show a 3-vertex example of a graph on which Dijkstra's algorithm always fails. Please clearly identify which vertex is the source. [3 marks]

3. Use Prim's algorithm starting at node A to compute the Minimum Spanning Tree (MST) of the following graph. Write down the edges of the MST in the order in which Prim's algorithm adds them to the MST. Use the format (node1; node2) to denote an edge. [4 marks]



**Question 4: Number Theory and Cryptography [20 marks]**

1. Compute $3^{201} \bmod 11$. Please indicate each step and how you arrive at each answer. [3 marks]
2. Compute $2^{802} \bmod 505$. Please indicate each step and how you arrive at each answer [3 marks]
3. Evaluate $\phi(1716)$ wherein $\phi$ is the Euler totient function. Please indicate each step and how you arrive at each answer. [3 marks]
4. Given RSA signatures X and Y for messages x and y, compute the signature of message x*y. (Assume there is no hashing: i.e. $X=x^d \bmod n$ for some unknown private exponent d and the public modulus is n). [5 marks]
5. Suppose, Amason is using RSA with modulus n and public exponent e. One day they are hacked, and their private key d becomes known to the attackers. Bob, the security consultant, suggests that instead of regenerating the new keys completely from the scratch, only the new exponents e', d' need to be re-computed, leaving the modulus n unchanged (after all, indeed modulus computation requires more work).
Is this safe? If yes, explain why. If not, show how the pirates can compromise the new system (i.e. compute new d' from e, d, n, e'). [6 marks]

**Question 5: NP-Hardness [30 marks]**

Consider the decision version of the Knapsack problem, and the subset sum problem.

**Knapsack**. The input is composed of two arrays s[1 . . . n], and v[1 . . . n] of positive integers, which specify the sizes and values of n items, respectively, a target value V , and a knapsack size S. The output is YES if it is possible to fit a subset of items of total value V in the knapsack, and it is NO, otherwise.

**Subset sum.** The input is a set X of n integers and a target integer t. The output is YES if there exists a subset of X, whose elements sum to t, and NO, otherwise.

Answer the following questions.

1. Describe a polynomial time reduction from subset sum to Knapsack. [7 marks]
2. Show that your reduction is correct; prove that the output of the Knapsack instance is YES if and only if the output of the subset sum instance is YES. [8 marks]
3. Explain why this reduction implies that Knapsack is NP-hard. [7 marks]
4. Prove that Knapsack is NP-Complete. [8 marks]

**END OF EXAM PAPER**