CSE317 Design & and Analysis of Algorithms

First Term Examination – Spring'15

Max Marks: 50 Time Allowed: 90 minutes

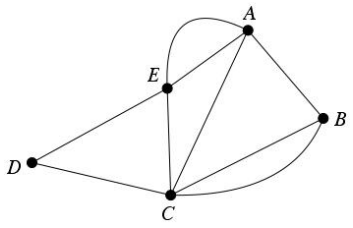
Attempt all questions. Be to the point. Show your work.

1. [10 marks] *3-way-Merge Sort:* Suppose that instead of dividing in half at each step of Merge Sort, you divide into thirds, sort each third, and finally combine all of them using a three-way merge subroutine. What is the overall asymptotic running time of this algorithm? Give a recurrence describing the running time of 3-way Merge Sort and solve it to justify your answer. (Hint: Note that the merge step can still be implemented in $O(n)$ time.)

2. [10 marks] Suppose we modify the deterministic linear-time selection algorithm by grouping the elements into groups of 7, rather than groups of 5. (Use the "median-of-medians" as the pivot, as before.) Does the algorithm still run in $O(n)$ time? (Give the recurrence for this case and solve it.)

3. Let $A[1..n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an *inversion* of A.
   (a) [2 marks] List the five inversions of the array $[2, 3, 8, 6, 1]$.
   (b) [3 marks] What array with elements from the set $\{1, 2, \ldots, n\}$ has the most inversions? How many does it have?
   (c) [5 marks] Run the $O(n \log n)$ algorithm from the lecture to count the number of inversion in array $A = [1, 5, 4, 8, 10, 2, 6, 9, 12, 11, 3, 7]$.

4. Consider the following graph.



   (a) [1 mark] What is the degree of vertex A?
   (b) [2 marks] Give an adjacency-list representation of the above graph.
   (c) [2 marks] What is the minimum number of edges crossing some cut in the above graph?
   (d) [5 marks] Use the Edge Contraction Algorithm to find a cut in this graph.

5. [10 marks] $A[1..n]$ is said to have a *majority* element if more than half of its entries are the same (so if $n = 6$ or $n = 7$, any majority element will occur in at least 4 positions). Given an array, the task is to design an efficient algorithm to tell whether the array has a majority element, and, if so, to find that element. The elements of the array are not necessarily from some ordered domain like the integers, and so there can be no comparisons of the form "is $A[i] > A[j]$?". (Think of the array elements as GIF files, say.) However you can answer questions of the form: "is $A[i] = A[j]$?" in constant time. Show how to solve this problem in $O(n \log n)$ time. (Hint: Split the array A into two arrays A_1 and A_2 of half the size. Does knowing the majority elements of A_1 and A_2 help you figure out the majority element of A? If so, you can use a divide-and-conquer approach.)