End-Semester Examination: Algorithms-II: CS31005: Autumn 2015

Maximum marks: 100: Time : 3 hours
Write precisely and with rigour.
You may make reasonable assumptions and justify them in your answers in case there is any need to do so.

Department of Computer Science and Engineering
IIT Kharagpur

(1) Show that the size of the minimum vertex cover in an undirected, simple and connected graph $G(V, E)$, cannot be smaller than the size of the maximum matching M in G. So, you may get a reasonably large sized subset $VC1$ of size $|M|$, of a vertex cover for G, by selecting an arbitrary vertex from each of the edges in M as a first step. Then, you may delete all edges of the graph covered by vertices in $VC1$, and find the maximum matching M' in the remaining graph G'. We can therefore add $|M'|$ more vertices (in this second step), to the $|M|$ vertices we already have from the first step. Suppose we keep doing this process of finding successive maximum matchings M, M', M'', until the remaining graph becomes empty. Finally, we will therefore have a vertex cover for G. Since computing these successive maximum matchings is possible in polynomial time, this algorithm for computing an approximate vertex cover is also a polynomial time algorithm. Compare the worst-case approximation ratio of this algorithm with that of the 2-factor vertex cover algorithm where we were required to only find a single *maximal* matching $M_{maximal}$, and a vertex cover of size $2|M_{maximal}|$. You are required to compare the approximation ratios for graphs such as paths and cycles (of even and odd lengths), cliques, and complete bipartite graphs, for these two heuristics as mentioned above. [10 marks]

(2) We have a *non-convex, multiply connected* piece of land P, with corners/vertices $p_0, p_1, \ldots p_{n-1}$. This piece of land P has several *holes or islands*. We also have a triangulation of the *interior of* P, which excludes the regions inside the holes. The interior of P is the region bounded by the *outer boundary* of P, and excludes the *interiors* of all the *holes or islands* of simple polygonal kind, where each such hole is defined by its own boundary. The smallest convex polygon C, that includes the whole of P is also given. The regions in $C \setminus P$, form some simple polygons belonging to the exterior of P inside C. Given a *query* point q, anywhere in the 2d plane, we wish to determine in which of the triangles of the triangulation of the interior of P, the point q lies in, provided it lies inside in the interior of the piece of land P. The query must be answered in $O(\log n)$-time. State how you would create a suitable data structure in $O(n \log n)$ time so that such queries can be answered very quickly, that is, in $O(\log n)$ time. Keep your answer brief and simply outline the main issues of your design. [5+5 marks]

(3) State the *general cutting lemma* for the two dimensional case, and the *weak cutting lemma* too.

For proving the weak cutting lemma, n lines are given and we have to select $s = 6n \ln r$ lines in a random sample for the desired $\frac{1}{r}$-cutting. Prove that there is indeed such a $\frac{1}{r}$-cutting. [3+3+4 marks]

(4) In the random sampling example where we select r points out of n points on a line, we know

that each of the $r+1$ intervals has at most $O(\frac{n}{r}\log r)$ points of the unselected $n-r$ points with probability at least $\frac{1}{2}$. Show that we can create an $(r+1)$-ary search tree using random sampling and recursion with logarithmic query time. Write the recurrence relations for search time and for the Las Vegas recursive creation of the 'balanced' search tree. [10 marks]

(5) Given a simple *monotone* polygon P of n vertices, show how one stack suffices to triangulate P in linear time. Analyze the time complexity after showing that your algorithm correctly computes a triangulation. [8 marks]

(6) Suppose a simple polygon P of n vertices is triangulated and the triangulation is provided as a graph $G(V, E)$, where V is the set of $n-2$ triangles and E is the set of pairs of triangles that share a triangulation edge. What is the cardinality of E? Is G a connected graph? Suppose we take a point p inside some triangle $v \in V$. Clearly the whole of the triangle v is *visible* from p. Since edges of triangle v are visible from p, some points in the triangles $w \in N(v)$ are also visible from p, where $N(v)$ denotes the neighbouring triangles of v in $G(V, E)$. State how you will compute the portions of such triangles $w \in N(v)$, that are visible from p. Since v has at most three neighbours in $N(v)$, we may need to determine portions of the remaining at least $n-6$ triangles that are visible from p. Design an algorithm for doing so. Analyze the running time of your algorithm. [15 marks]

(7) Show that the *edge collapsing Monte Carlo* algorithm for computing a *minimum cardinality* cut in a connected, undirected graph, indeed yields a specific minimum cut with probability at least $\frac{1}{\binom{n}{2}}$. Using this result, show that in an n-vertex connected, undirected graph, the number of distinct *minimum cardinality* cuts is at most $\binom{n}{2}$. [6+4 marks]

(8) We know that the discrepancy of an n-vertex hypergraph $G(V, E)$ with m hyperedges is $O(\sqrt{2n\ln 2m})$. We also know that there is a Las Vegas (expected polynomial running time) algorithm for determining a bicoloring χ of the vertices of G, such that the discrepancy (or imbalance) in each hyperedge e under bicoloring χ is at most $\sqrt{3|e|\ln 2m}$. We wish to study the expected running time for computing a bicoloring χ', where the discrepancy in each hyperedge e under bicoloring χ' is at most $\sqrt{c|e|\ln 2m}$, for any $c > 2$. Analyze how the expected running time will change when the value of c is reduced below 3. You may analyze for $c = 2.5$. [10 marks]

(9) We know that the DFT of the coefficient vector of a polynomial can be computed in $O(n\log n)$ time using the FFT algorithm, where we are dealing with polynomials of degree $n-1$. How does this facilitate the computation of the product polynomial of two $(n-1)$-degree polynomials in $O(n\log n)$ time? [5 marks]

(10) In the Miller-Rabin primality testing algorithm, in addition to Fermat's theorem, we also use the modular exponentiation method and a property of composite numbers. State this property and show in a brief description how the exponentiation is done efficiently. [4+3 marks]

(11) Briefly state the random sampling step in the *randomized selection algorithm* for sampling the *multiset* R of *sublinear* size with respect to the input S of n unsorted elements. Argue how the sorting steps in the algorithm have $o(n)$ comparisons. If we wish to determine the element $S_{(k)}$, the kth smallest element in S, then state how the element $S_{(k)}$ is picked up from a suitable $o(n)$ sized subset P of S. State the total running time of one iteration of this method, and the probability of failure in determining $S_{(k)}$, without stating the analysis. [5 marks]

==