Mid-semester Examination: Autumn 2011: CS60007: Algorithms Design and Analysis

Department of Computer Science and Engineering, IIT Kharagpur
LTP 4-0-0: Credits 4: Time 2 hours: Marks 100
There is limited choice: Maximum marks 100

1. Suppose we perform independent random bicoluring of the n vertices in the vertex set V of an n-vertex r-uniform hypergraph $G(V, E)$ with the set E of hyperedges. Each vertex gets either colour, 0 or 1, with equal probability.

   (a) Based on the number $|E|$ of hyperedges, estimate an upper bound on the probability that the random bicolouring does not properly bicolour some hyperedge in E.

   (b) If $r = \log_2 n$ then determine an upper bound on the number $|E|$ of hyperedges so that $G(V, E)$ is *properly bicolourable*, that is, there is some bicoloring of vertices in V which renders every hyperedge in E non-monochromatic.

   (c) Can this bound give rise to a Las Vegas bicoloring algorithm for $G(V, E)$ with polynomial expected running time? If not, further modify/improve your upper bound on the number of hyperedges by reducing the number of hyperedges, in order to do so.

   (d) What happens to the upper bound in (c) above when $r = (\log_2 n)^k$, where $k > 1$ is a positive natural number?

   (e) What happens to the upper bound in (d) above when $k = \log_2 n$?

   [5+5+7+4+4 marks]

2. Define *discrepancy* for a hypergraph $G(V, E)$ and show that the discrepancy of an n-vertex r-uniform hypergraph is $O(\sqrt{n \log n})$ if r is a constant with respect to asymptotic n. What happens to this bound when $r = \log_2 n$? [5+10+5 marks]

3. Explain the main principles of the method of *conditional expectations* in the *derandomizing* process where a deterministic decision making step replaces a random step. Illustrate using the example of bicolouring vertices of a hypergraph for bounded discrepancy bicoloring in order to show how the sequence of conditional expectations are enforced to remain upper bounded by a small fraction. [10 marks]

4. Estimate the cardinality/size of the $\frac{1}{r}$-net created by the *greedy method* for a set system (hypergraph) $G(V, E)$ with a set V of n vertices and a set E of m subsets of V (hyperedges)? [10 marks]

5. Consider the greedy algorithm for the *weighted set cover* problem, where the next set S to be included in the set cover is one that has the minimum *cost effectiveness* $\alpha(S)$. *Cost effectiveness* $\alpha(S)$ is defined as $\frac{c(S)}{|S \setminus C|}$, where C is the set of elements already covered by previously selected sets, and $c(S)$ is the positive weight of set S. We also define $price(e) = \alpha(S)$ for each element $e \in S \setminus C$. Note that $C \cap S$ may not be empty. Let $e_i, 1 \leq i \leq k$ be the ith element of S to be covered in the algorithm. Here, $|S| = k$. Then, show that $price(e_i) \leq \frac{c(S)}{k-i+1}$. [10 marks]

6. State the method of rounding the solutions of a linear program in order to compute an *approximate weighted vertex cover* of an undirected vertex-weighted graph? Establish the upper bound on the approximation ratio in the algorithm. [Here, all the vertex weights are positive rational numbers. The weight of a vertex cover is the sum of weights of its vertices.] [5+10 marks]

7. Let $G(V, E)$ be an undirected graph. Let $M \subset E$ be such that for each pair (e, f) of distinct edges e and f of M, there is no vertex common to e and f. Answer the following questions.

   (a) If VC is a vertex cover for $G(V, E)$ then is $|VC| > |M|$? Why?

   (b) Does the set of all vertices of edges in M constitute a vertex cover for $G(V, E)$? Why?

   [5+5 marks]

8. Let $N(V, E, s, t, c)$ be a *network* with vertex set V, edge set E, *source* s, *sink* t and *capacity* function c, where $c(u, v) \geq 0$ is the capacity of the directed edge $(u, v) \in E$. Let f^* be the flow function with maximum flow from s to t in network N, denoted by $|f^*|$. Design a linear time algorithm for computing a *minimum capacity* cut (S, T), where $s \in S$, $t \in T$ and $S \cap T = \phi$. and $S \cup T = V$. [10 marks]

9. Suppose we repeatedly augment flows from the source s to the sink t in residual networks along paths with maximum bottleneck capacity. Then show that the the maximum flow f^* can be computed in $O(m \log |f^*|)$ augmentations, where m is the number of edges of the network. [Hint: Assume that the maximum flow $|f^*|$ is the sum of a set of m augmenting flows.] [10 marks]