INTRO TO APPROXIMATION, CLASS 7

Leftovers: 3-wise independent variables, parallel algorithms, and matching

3-wise independent bits

From the last time:

D: A set of random variables $X_1, \ldots X_n$ is *k-wise independent* if for any subset $I \subseteq \{1, \ldots, n\}$ with $|I| \leq k$ and any possible outcome values c_i, the multiplication property for independence holds:

$$Pr[\bigwedge_{i \in I}(X_i = c_i)] = \prod_{i \in I} Pr[X_i = c_i].$$

Exercise one We know that k bits are sufficient for generating $O(2^k)$ many random pairwise-independent variables. The question now is: how many do we need for 3-wise independent variables?

Surprisingly, you can generate 2^{k-1} of them using again just k bits. Suggest a generator and prove that the result are 3-wise independent random bits.

Parallel algorithms

Exercise two Design a deterministic parallel algorithm, which, given a graph $G = (V, E)$ and some subset $X \subseteq V$, is able to determine whether X is a inclusion-wise maximal independent set in time $O(\log |E|)$ and with $O(|E|)$ processors.

Exercise three An *inclusion-wise maximal matching* in a graph is any matching which cannot be improved by just adding an edge (without any removals).

Design a parallel randomized Las Vegas algorithm which can find such a matching.

Exercise four Design a parallel randomized Las Vegas algorithm which generates a uniformly random permutation on n elements. This will be a very different algorithm compared to the ones we have seen at the lecture, so let us break the task into steps:

a) One possible solution is built upon injective functions. Suppose we have some set $X = \{1, \ldots n\}$ and $Y = \{1, \ldots, m\}$, $m \geq n$. What is the probability that a uniformly random function $f : X \to Y$ is injective?
b) Suppose that I give you a uniformly random injective function $f : X \to Y$ at the start of the algorithm. Can you create a uniformly random permutation out of it?
c) Can you quickly test in parallel that a given function $f : X \to Y$ is injective?
d) Can you now generate a uniformly random permutation using the above?

Perfect matchings

EXERCISE FIVE Prove that the rank of the Edmonds matrix of a bipartite graph G is equal to the size of the largest matching in G.

The Edmonds matrix of a bipartite graph $G = (U, V, E)$ with partites $U = \{u_1, \ldots, u_n\}$ and $V = \{v_1, \ldots, v_n\}$ is defined as a matrix B of polynomials of size $n \times n$ such that

$$B_{ij} = \begin{cases} x_{ij} & \text{if } (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

A similar result holds for general graphs: the rank of the Tutte matrix of a graph G is equal to two times the size of the largest matching in G.

The Tutte matrix of a graph $G = (V, E)$ with n vertices is defined as a matrix T of polynomials of size $n \times n$ such that

$$B_{ij} = \begin{cases} x_{ij} & \text{if } (v_i, v_j) \in E, i < j \\ -x_{ji} & \text{if } (v_i, v_j) \in E, i > j \\ 0 & \text{otherwise} \end{cases}$$

That is, $T_{ij} = x_{ij}$ and $T_{ji} = -x_{ij}$ for $i < j$.