CMPT 705 — Design and Analysis of Algorithms

Exercises on PSPACE and Randomization. Due: Tuesday, November 15th (at the beginning of the class)

Reminder: the work you submit must be your own. Any collaboration and consulting outside resources must be explicitly mentioned on your submission.

1. Let us consider a special case of Quantified 3-SAT in which the underlying Boolean formula has no negated variables. Specifically, let $\Phi(x_1, \ldots, x_n)$ be a Boolean formula of the form $C_1 \wedge \ldots \wedge C_k$, where each C_i is a disjunction of three terms. We say Φ is *monotone* if each term in each clause consists of a nonnegated variable, that is, each term is equal to x_i, for some i rather than $\neg x_i$.

   We define Monotone QSAT to be the decision problem

   $$\exists x_1 \forall x_2 \ldots \exists x_{n-2} \forall x_{n-1} \exists x_n \Phi(x_1, \ldots, x_n)?$$

   where the formula Φ is monotone.

   Do one of the following two things: (a) prove that Monotone QSAT is PSPACE-complete by reducing Quantified 3-SAT to it; or (b) give an algorithm to solve an arbitrary instance of Monotone QSAT that runs in time polynomial in n.

2. Prove that deciding if the first player in the Geography game on graphs (see lecture slides) has a forced win is PSPACE-complete.

3. Let $G = (V, E)$ be an undirected graph with n nodes and m edges. For a subset $X \subseteq V$, we use $G[X]$ to denote the subgraph induced on X — that is the graph whose node set is X and whose edge set consists of all edges of G for which both ends lie in X.

   We are given a natural number $k \leq n$ and are interested in finding a set of k nodes that induces a 'dense' subgraph of G; we will phrase this concretely as follows. Give a polynomial time algorithm that produces, for a given natural number $k \leq n$, a set $X \subseteq V$ of k nodes with the property that the induced subgraph $G[X]$ has at least $\frac{mk(k-1)}{n(n-1)}$ edges.

   You may give either (a) a deterministic algorithm, or (b) a randomized algorithm that has an expected running time that is polynomial, and that only outputs correct answers.

4. Suppose we have a system with n processes. Certain pairs of processes are in conflict, meaning that they both require access to a shared resource. In a given time interval, the goal is to schedule a large subset S of the processes to run — the rest will remain idle — so that no two conflicting processes are both in the scheduled set S. We will call such a set S conflict-free.

   One can picture this process in terms of a graph $G = (V, E)$ with a node representing each process and an edge joining pairs of processes that are in conflict. It is easy to check that a set of processes is conflict-free if and only if it forms an independent set in G. This suggests that finding a maximum size conflict-free set S, for an arbitrary graph G, will be difficult (since the general Independent Set problem is reducible to this problem). Nevertheless, we can still look for heuristics that find a reasonably large conflict-free set. Moreover, we would like a simple method for achieving this without centralized control: Each process should communicate with only a small number of other processes and then decide whether or not it should belong to the set S.

We will suppose for purposes of this question that each node has exactly d neighbors in the graph G. (That is, each process is in conflict with exactly d other processes.)

(a) Consider the following simple protocol.

Each process P_i independently picks a random value x_i; it sets x_i to 1 with probability $\frac{1}{2}$ and sets x_i to 0 with probability $\frac{1}{2}$. It then decides to enter the set S if and only if it chooses the value 1, and each of the processes with which it is in conflict chooses the value 0.

Prove that the set S resulting from the execution of this protocol is conflict free. Also give a formula for the expected size of S in terms of n (the number of processes) and d (the number of conflicts per process).

(b) The choice of the probability $\frac{1}{2}$ in the protocol above was fairly arbitrary, and it is not clear that it should give the best system performance. A more general specification of the protocol would replace the probability $\frac{1}{2}$ by a parameter p between 0 and 1, as follows.

Each process P_i independently picks a random value x_i; it sets x_i to 1 with probability p and sets x_i to 0 with probability $1 - p$. It then decides to enter the set S if and only if it chooses the value 1, and each of the processes with which it is in conflict chooses the value 0.

In terms of the parameters of the graph G, give a value of p so that the expected size of the resulting set S is as large as possible. Give a formula for the expected size of S when p is set to this optimal value.