HW1 Randomized Algorithms

MPRI 1.24

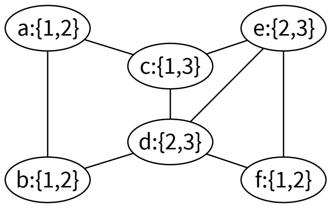
Tue. Nov. 18, 2014 - **Due on Tue. Nov. 25, 2014**

You are asked to complete the exercise marked with a [★] and to send me your solutions at:
`nicolas.schabanel@liafa.fr`
(or drop it in my mail box at the 4th floor of Sophie Germain) on **Tue. Nov. 25, 2014**.

■ **Exercise 1 (Constraint-based 3-coloring randomized algorithm).** Given an undirected graph $G = (V, E)$ with a pair p_u of integers chosen in $\{1, 2, 3\}$ on each vertex u, the *constrained 3-coloring* problem consists in finding a valid 3-coloring α of the vertices such that the color of each vertex u is chosen in p_u, i.e. an $\alpha : V \to \{1, 2, 3\}$ such that:

$$\text{for all } uv \in E,\ \alpha_u \neq \alpha_v \quad \text{and} \quad \text{forall } u \in V, \alpha_u \in p_u.$$

► ***Question* 1.1*)*** *Give a solution to the following constrained 3-coloring instance:*



► ***Question* 1.2*)*** *Show that solving a constrained 3-coloring instance is equivalent to solve a 2-SAT instance that you are asked to describe explicitly. Give the 2-SAT instance corresponding the graph above.*

We consider the following algorithm for 3-coloring an undirected graph $G = (V, E)$.

Algorithm 1 Constraint-based 3-coloring randomized algorithm

---

repeat
- Choose independently for each vertex u a pair p_u of integers in $\{1, 2, 3\}$ uniformly at random.
- Solve the corresponding constrained 3-coloring using the reduction above to a 2-SAT instance.

until a 3-coloring is found

► ***Question* 1.3*)*** *Show that if G is 3-colorable, then this algorithm will find a 3-coloring after at most $O((\frac{3}{2})^n)$ iterations on expectation, where $n = |V|$. Give an upper bound on the expected total computation time of this algorithm if this case.*

► ***Question* 1.4*)*** *After how many iterations, should we stop the algorithm and declare the graph as "not 3-colorable" so as to be mistaken with probability at most $\frac{1}{n^2}$? What is the failure probability if the graph is indeed not 3-colorable?*

■ **Exercise 2 (Better randomized algorithm for Min Cut).** [★]

Consider the following recursive algorithm for Min Cut:

Procedure FastMinCut(G : an undirected multigraph)
if $n \leqslant 6$ **then**
  return an optimal min-cut obtained by exhaustive search.
else
  $t := 1 + \lceil n/\sqrt{2} \rceil$.
  Make two copies G_1 and G_2 of G.
  for $i = 1..2$ **do**
    Make $n-t$ independent random edge contractions on G_i to obtain multigraph H_i with t vertices.
  return the best cut between FastMinCut(H_1) and FastMinCut(H_2).

▶ ***Question 2.1)*** *Show that its running time is $T(n) = O(n^2 \log n)$.*

▶ ***Question 2.2)*** *Show that it uses at most $M(n) = O(n^2)$ memory.*

▶ ***Question 2.3)*** *Show that the probability that a min cut survives $n-t$ random edge contractions is at least $\frac{1}{2}$ when $n > 6$.*

Let $P(k)$ be the minimum probability that the algorithm outputs a min cut for a multi-graph that requires k levels of recursions ($k = \Theta(\log n)$ if G has n vertices).

▶ ***Question 2.4)*** *Show that $P(k) \geqslant p(k)$ where $p(0) = 1$ and $p(k+1) = p(k) - \dfrac{p(k)^2}{4}$.*

Let $q(k) = \dfrac{4}{p(k)} - 1$ so that $q(0) = 3$ and $q(k+1) = q(k) + 1 + \dfrac{1}{q(k)}$.

▶ ***Question 2.5)*** *Show that for all $k \geqslant 0$, $k < q(k) \leqslant k + H_{k-1} + 3$, where $H_k = \sum_{i=1}^{k} \frac{1}{i}$ is the kth harmonic number.*

▶ ***Question 2.6)*** *Conclude that* FastMinCut *computes a min cut with probability $\Omega(1/\log n)$. Propose an algorithm which increases the success probability to $1 - 1/n^2$. Compare its time complexity to the best known deterministic algorithm (based on a max flow computation), $O(mn \log(m^2/n))$.*