CS 473 ✧ Spring 2017

Homework 11

"Due" Wednesday, May 3, 2017 at 8pm

This homework will not be graded.
However, material covered by this homework *may* appear on the final exam.

---

1. Let Φ be a boolean formula in conjunctive normal form, with exactly three literals in each clause. Recall that an assignment of boolean values to the variables in Φ ***satisfies*** a clause if at least one of its literals is TRUE. The ***maximum satisfiability problem*** for 3CNF formulas, usually called MAX3SAT, asks for the maximum number of clauses that can be simultaneously satisfied by a single assignment.

   Solving MAX3SAT exactly is clearly also NP-hard; this question asks about approximation algorithms. Let $\mathit{Max3Sat}(\Phi)$ denote the maximum number of clauses in Φ that can be simultaneously satisfied by one variable assignment.

   (a) Suppose we assign variables in Φ to be TRUE or FALSE using independent fair coin flips. Prove that the expected number of satisfied clauses is at least $\frac{7}{8}\mathit{Max3Sat}(\Phi)$.

   (b) Let k^+ denote the number of clauses satisfied by setting every variable in Φ to TRUE, and let k^- denote the number of clauses satisfied by setting every variable in Φ to FALSE. Prove that $\max\{k^+, k^-\} \geq \mathit{Max3Sat}(\Phi)/2$.

   (c) Let $\mathit{Min3Unsat}(\Phi)$ denote the *minimum* number of clauses that can be simultaneously left *unsatisfied* by a single assignment. Prove that it is NP-hard to approximate $\mathit{Min3Unsat}(\Phi)$ within a factor of $10^{10^{100}}$.

2. Consider the following algorithm for approximating the minimum vertex cover of a connected graph G: ***Return the set of all non-leaf nodes of an arbitrary depth-first spanning tree***. (Recall that a depth-first spanning tree is a rooted tree; the root is not considered a leaf, even if it has only one neighbor in the tree.)

   (a) Prove that this algorithm returns a vertex cover of G.

   (b) Prove that this algorithm returns a 2-approximation to the smallest vertex cover of G.

   (c) Describe an infinite family of connected graphs for which this algorithm returns a vertex cover of size *exactly* $2 \cdot \mathrm{OPT}$. This family implies that the analysis in part (b) is tight. *[Hint: First find just **one** such graph, with few vertices.]*

3. Consider the following modification of the "dumb" 2-approximation algorithm for minimum vertex cover that we saw in class. The only change is that we return a set of edges instead of a set of vertices.

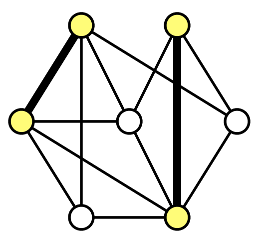
```
ApproxMinMaxMatching(G):
  M ← ∅
  while G has at least one edge
      uv ← any edge in G
      G ← G \ {u, v}
      M ← M ∪ {uv}
  return M
```

   (a) Prove that the output subgraph M is a *matching*—no pair of edges in M share a common vertex.

   (b) Prove that M is a *maximal* matching—M is not a proper subgraph of another matching in G.

   (c) Prove that M contains at most twice as many edges as the *smallest* maximal matching in G.

   (d) Describe an infinite family of graphs G such that the matching returned by ApproxMinMaxMatching(G) contains exactly twice as many edges as the smallest maximum matching in G. This family implies that the analysis in part (c) is tight. *[Hint: First find just* ***one*** *such graph, with few vertices.]*



The smallest maximal matching in a graph.