CS 466/666, Spring 2009
Timothy Chan

Assignment 4 (due July 28 Tuesday 5pm)

Please read `http://www.student.cs.uwaterloo.ca/~cs466/policies.html` first for general instructions.

1. [*18 marks*] Consider the following job scheduling problem: we have $n/3$ processors available, each capable of handling 3 jobs; we would like to assign jobs to processors so as to minimize the overall (maximum) completion time. More precisely, given a sequence of n positive numbers $s_1, \ldots, s_n$ (where n is divisible by 3), we would like to partition the sequence into subsets $S_1, \ldots, S_{n/3}$, each containing 3 numbers, so as to minimize the quantity $\max_{j=1,\ldots,n/3} \sum_{s \in S_j} s$.

   (a) [*4 marks*] Show that there is an *online* approximation algorithm with approximation factor at most 3. [Hint: the algorithm is *really* simple. For the analysis, define $M = \max\{s_1, \ldots, s_n\}$ and compare with the optimal value. . .]

   (b) [*4 marks*] Let k be a fixed constant and let A be a fixed set of k elements. Show that in the special case where all s_i's come from the set A (duplicates are allowed), the (offline) problem can be solved exactly in polynomial time.

   (c) [*10 marks*] Now design and analyze a polynomial-time approximation scheme (PTAS) for the general (offline) problem. [Hint: use a rounding technique (but it should be simpler than the bin-packing PTAS from class).]

2. [*17 marks*] We have mentioned in class that the maximum independent set problem is hard to approximate in general. In this question, you will investigate approximation algorithms for maximum independent set in a special case: sparse graphs. Suppose the input graph $G = (V, E)$ has n vertices and m edges with $m \leq cn$ for some integer constant c.

   (a) [*8 marks*] Consider the following greedy algorithm:

   ```
   1. S = ∅
   2. while G is not empty do {
   3.     pick a vertex v of the lowest degree
   4.     insert v to S, and remove v and its neighbors from G
      }
   5. return S
   ```

   Prove that the algorithm returns a feasible solution in polynomial time and has approximation factor $1/(8c)$.

   [Hint: Let V_L be the set of vertices with degree less than $4c$. First show that $|V_L| \geq n/2$. Then show that at least $|V_L|/(4c)$ vertices of V_L are chosen in S.]

(b) [*9 marks*] Consider the following randomized algorithm:

1. let π be a random ordering of V
2. for each $v \in V$ do
3. put v in S iff all neighbors u of v appear after v in the ordering π

Prove that the algorithm returns a feasible solution in polynomial time and has expected approximation factor $1/(2c+1)$.

[Hint: Given $d+1$ fixed elements $v, v_1, \ldots, v_d$, what is the probability that v appears before $v_1, \ldots, v_d$ in a random permutation? Prove that the expected size of S is at least $\sum_{v \in V} 1/(\deg(v)+1)$. To conclude, you may use the following inequality without proof: $\sum_{i=1}^{n}(1/a_i) \geq n^2 / \sum_{i=1}^{n} a_i$ (which follows from a known inequality about the "arithmetic mean" and "harmonic mean").]