Homework 3
Due 2004/12/13

Oded Regev
Department of Computer Science
Tel Aviv University

1. Find a basis $b_1, \ldots, b_n$ such that after we apply one reduction step of the LLL algorithm to it, the maximum length of a vector in it *increases* (even by as much as $\Omega(\sqrt{n})$).

2. Using the LLL algorithm, construct an algorithm for solving SVP exactly in time $2^{O(n^2)} \cdot \mathrm{poly}(D)$ where n is the rank of the lattice and D is the input size. Hint: show that if we represent the shortest vector as an integer combination of the LLL-reduced basis, none of the coefficients can be larger than 2^{cn} for some c.

3. Show that our analysis of the LLL algorithm is tight (perhaps up to some constant) by demonstrating a 'bad' basis. For simplicity, show this only for $\delta = \frac{3}{4}$.

4. Improve the bound B in Coppersmith's algorithm to $N^{1/d}$. Hint: for some $h \geq 2$, consider the set of polynomials $\{N^{h-i-1} f(x)^i x^j \mid 0 \leq j < d,\ 0 \leq i < h\}$ and work modulo N^{h-1}. For $h = 2$ this is the set we had in class. Show that we can obtain $B = N^{1/d-\varepsilon}$ for any constant $\varepsilon > 0$ by setting h to be a large enough constant. Next show that by setting h to be some function of N, this becomes $B = N^{1/d}/C$ for some constant $C > 0$. Finally, show how to obtain $B = N^{1/d}$ using this.

5. For any $\gamma \geq 1$, describe a randomized Karp reduction from GapSVP_γ to GapCVP_γ. The reduction should map NO instances to NO instances with probability 1 and YES instance to YES instances with some probability $p > 0$. First show this for $p = \frac{1}{n}$ and later try to improve this to $p = \frac{1}{2}$. Hint for $p = \frac{1}{2}$: given an SVP instance $(b_1, \ldots, b_n)$ construct the CVP instance $(2b_1, b_2 + c_2 b_1, \ldots, b_n + c_n b_1)$ with the target vector b_1 where $c_2, \ldots, c_n$ are independently chosen to be 0 or 1 with probability $\frac{1}{2}$.

6. Prove that $\mathsf{GapSVP}_{\sqrt{n}} \in \mathsf{NP} \cap \mathsf{coAM}$ (try to find an AM protocol similar to the one we had in class and not use the 'large cube' approach).