

**COT 5405: Advanced Algorithms**  
**Fall 2012**

**Assignment 2**

**Due: 5pm, 30 Nov 2012**

1. (20 points) Consider the following instance of MAX-SAT.  $C = (x_1 \vee x_2) \wedge (x_1 \vee x_3 \vee x_4) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_3)$  with weights 2, 2, 1, and 3 respectively. Use the derandomized algorithm of section 16.2 to find an approximate solution. Give a good a-posteriori bound for your solution. Show all steps.
2. (20 points) Assume that arrays  $A[n][n]$  and  $B[n][n]$  are stored in row major order. Give a  $\Theta(n^2)$  cache-complexity algorithm to store the transpose of A in B. Then give a  $\Theta(n^2/L)$  cache-aware algorithm.
3. (20 points) Consider the following linear recurrence:  $x_{i+1} = a_i + b_i x_i$ , where  $x_0$  is given. Formulate the solution of this recurrence as a prefix computation and give the parallel time complexity, speedup, and efficiency if parallel prefix is used to solve it.
4. (20 points) Give a strict quadratic program for the MAX k-CUT problem (defined in exercise 2.3). Relax it to a vector program and given an equivalent semi-definite program for it.
5. (20 points) Consider any NP hard problem mentioned in <http://www.nada.kth.se/~viggo/problemlist/compendium.html>, which has a practical application. Collect realistic data for the problem and use a good approximation algorithm to solve it. Also solve it using some greedy heuristic that appears intuitively appealing and compare the performance against the approximation algorithm. If possible, determine good a-posteriori bounds for your solution. Give a short description of the problem, any important information on the problem mentioned in the compendium above, description of the approximation algorithm and the greedy heuristic, and performance results (time and quality of solution) for both algorithms.