Algorithms and Data Structures II, 2007
Fourth Assignment

1 Instructions

1. Below you find three problems. Solve the problem(s) with the number(s) corresponding to the seminar(s) you did not attend. Consequently, if you attended all three seminars, you do not have to do this assignment.

2. The due date is **November 12th**.

3. Submit as usual, by putting it in my postbox, house 1, floor 4.

4. Use the appropriate cover sheet.

5. Any questions are answered by e-mail: `jonathan.morndal@it.uu.se`.

Problems to hand in

1. **The Knapsack Problem.** You plan to go hiking and you want to pack your stuff in a knapsack. Your knapsack has a total volume of V cubic inches, and you have n different items of volumes $v_1, \ldots, v_n$ that you want to take with you. Unfortunately, you can't take them all, since their total volume exceeds the volume of your knapsack. You need to decide which items to bring and which items to leave at home.

   Suppose that you want to optimize the packing so that the volume is is utilized to its maximal extent. This means you want to find a subset of the items such that the sum of their volumes is maximal with the constraint that the sum is no more than V. We call this the optimum packing of $v_1, \ldots, v_n$ in V.

(i) Fix a list of items $v_1, \ldots, v_n$. For any $i \leq n$, and any $U \leq V$, denote by $\mathrm{OPT}_i(U)$ the volume of the optimum packing of $v_1 \ldots, v_i$ in U. Explain why we have, for all i and U, that

$$\mathrm{OPT}_i(U) = \max\{\mathrm{OPT}_{i-1}(U), v_i + \mathrm{OPT}_{i-1}(U - v_i)\}$$

(ii) Assume that all the volumes $v_1, \ldots, v_n$ are integral numbers of units. Design a bottom up dynamic-programming algorithm for solving the knapsack problem in this case. Analyze your algorithm.

2. Solve problem 16-2, p 402 in CLRS.

3. **(BFS)** Describe an algorithm that uses BFS to determine whether or not a given undirected graph is bipartite[1].

(i) Prove correctness of your algorithm.

(ii) Analyze running time of your algorithm.

[1] i.e. it is possible to colour it's verticies in such a manner that two adjacent nodes have different color, and only two colours are used for all nodes.