Tutorial 6

Proof sketches.

1. Consider the following algorithm:

Assume without loss of generality that the graph is connected (otherwise consider each component separately).

A. Color 0 for one of the vertices.

B. Loop:

If there is a vertex v which is not colored and has two neighbours already colored differently, then output: not 2-colorable and stop.

Else, if there is a vertex v which is not colored and has at least one neighbour w already colored then choose one such v, w, and set $color(v) = 1 - color(w)$.

End Loop

C. Output 2-colorable.

Now the above loop will either stop due to conflict as mentioned above, or will color all the vertices in a valid way. Note that it cannot be the case that some vertex is not colored at the end of the above loop (unless the algorithm stops due to output "not 2-colorable", as the graph is connected.

Q2: It is easy to see that Knapsack problem is in NP: just guess the subset S' of S and verify that its weight is $\leq K$ and value is $\geq V$.

To show that it is NP-complete, reduce from partition:

If A is the set with s being the weight function in partition,

then make a knapsack problem as follows:

$S = A$. $value(a) = weight(a) = s(a)$, for each $a \in A$.

$K = \lfloor \frac{1}{2} * \sum_{a \in A} s(a) \rfloor$.

$V = \lceil \frac{1}{2} * \sum_{a \in A} s(a) \rceil$.

Now, it is easy to verify that there is a partition $A', A - A'$ in the partition problem iff the choice $A' \subseteq S$ satisfies the requirements of the knapsack problem.