NP and P problems: CHAPTER 34

34.1-4
Is the dynamic-programming algorithm for the 0-1 knapsack problem that is asked for in Exercise 15.2-2 a polynomial-time algorithm? Explain your answer.

15.2-2
Give a dynamic-programming solution to the 0-1 knapsack problem that runs in $O(nW)$ time, where n is the number of items and W is the maximum weight of items that the thief can put in the knapsack.

→ The $O(nW)$ algorithm runs in time that is linear W itself, not in $\log W$. If W is large, the value of W could be exponentially greater that the lenght of its binary representation.
For example, if W is on the order 2^m, its binary representation has lenght m. The running time $O(nW) = O(n \cdot 2^m)$ is exponential in m. Because of this it is considered a pseudo-polynomial time algorithm. It is polynomial in the numerical value of the input but not polynomial in the size of the input's encoding.

This isn't a polynomial-time algorithm. Recall that the algorithm from Exercise 16.2-2 had running time $\Theta(nW)$ where W was the maximum weight supported by the knapsack. Consider an encoding of the problem. There is a polynomial encoding of each item by giving the binary representation of its index, worth, and weight, represented as some binary string of length $a = \Omega(n)$. We then encode W, in polynomial time. This will have length $\Theta(\lg W) = b$. The solution to this problem of length $a + b$ is found in time $\Theta(nW) = \Theta(a \cdot 2^b)$. Thus, the algorithm is actually exponential.

34.2-2
Prove that if G is an undirected bipartite graph with an odd number of vertices, then G is nonhamiltonian.

→ Let $G = (V, E)$ be an undirected bipartite graph. By definition we know that for the disjoint subsets U and W, there exist a partition of its vertex such that

$$U \cup W = V \quad \text{and} \quad U \cap W = \emptyset$$

→ Consider any cycle C in G. Because the graph is bipartite, edges must alternate between vertices in U and vertices in W. This alternation enforces that any cycle must have the same number of vertices from U as from W.

→ Let's prove by contradiction that G has a Hamilton cycle C. The total lenght of the cycle would be $|V|$, the total number of vertices.

→ As the problem states we are going to suppose that for a bipartite graph $G = (V, E)$, and $|V| = |U| + |V|$ is odd, and there is Hamiltonian cycle. Now let's state that any cycle in a bipartite graph must have even lenght.

Proof: Let $G = (V, E)$, with vertices $U \cup W = V$, since the graph is bipartide let's label the nodes in a general way, that is $v_1 \in U$, and because every edge in a bipartite graph runs between the sets U and W, the neighbors of any vertex in U must lie in W, that is then $v_2 \in W$. Then, $(v_1, v_2) \in E$, then the next adjacent node to v_2, $v_3 \in U$, and so on. Thus, vertices $(v_1, v_3, v_5 \ldots)$ are in U

odd positions

even positions

and $(v_2, v_4, v_6 \ldots)$ are in W.

→ We continue our assumption. We start from v_1 and continue traveling through all the remaing vertices until we arrive to a node v_k which is the last node before coming back to v_1 and finish the cycle, since $|V| = |U| + |W|$ is odd this v_k is odd and must lie in U, contradiction occurs, there is no a cycle since after v_k we must go to W first!

34.2-6

A ***hamiltonian path*** in a graph is a simple path that visits every vertex exactly once. Show that the language HAM-PATH $= \{\langle G, u, v \rangle :$ there is a hamiltonian path from u to v in graph $G\}$ belongs to NP.

34.2-7

Show that the hamiltonian-path problem from Exercise 34.2-6 can be solved in polynomial time on directed acyclic graphs. Give an efficient algorithm for the problem.

CHAPTER 35

Approximation Algorithms