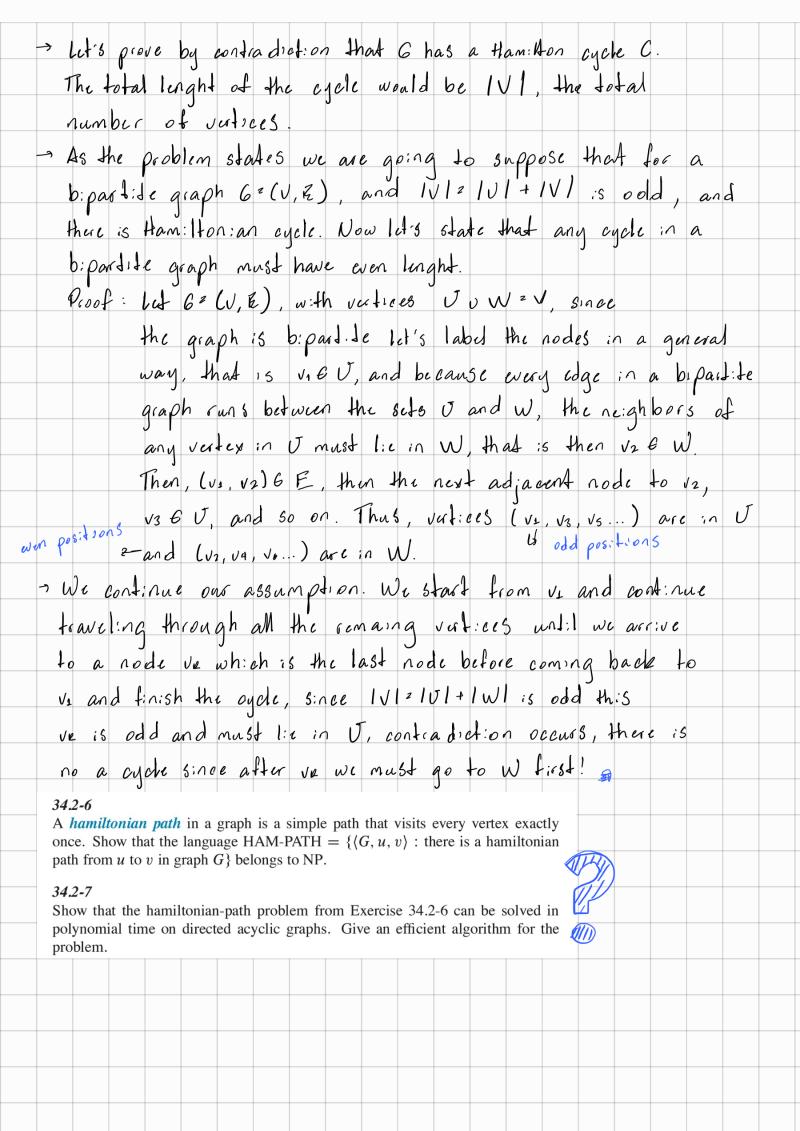
NP and P problems: CHAPTER 34														
34.1-4 Is the dynamic-programming algorithm for the 0-1 knapsack problem that is asked for in Exercise 15.2-2 a polynomial-time algorithm? Explain your answer.														
Give a dynamic-programming solution to the 0-1 knapsack problem that runs in $O(n \ W)$ 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 logw. If Wis large, the value of w could be exponentially														
greater that the length of its binary representation.														
For example, if wis on the order 2m, its binary representation														
has length m. The running time O(nw) = O(n.2m):6 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 bipastite graph. By definition														
we know that for the disjoint subsets U and W, there														
UNW 2 V and UnW 2 P														
- Consider any cycle Cin G. Because the graph is bipartite, edges														
must alternate between vulices in U and vutices in W. This atternation														
enforces that any cycle must have the same number of vertices from														
Dag from W.														



(	CHA	APT	ER	35													
					Ap	000	xim	af: e	20	Alo	00:	Jhn	15				