CSCI 570 Spring 2015 Discussion 13

1: Some **NP**-complete problems are polynomial-time solvable on special types of graphs, such as bipartite graphs. Others are still **NP**-complete.
Show that the problem of finding a Hamiltonian Cycle in a bipartite graph is still **NP**-complete.

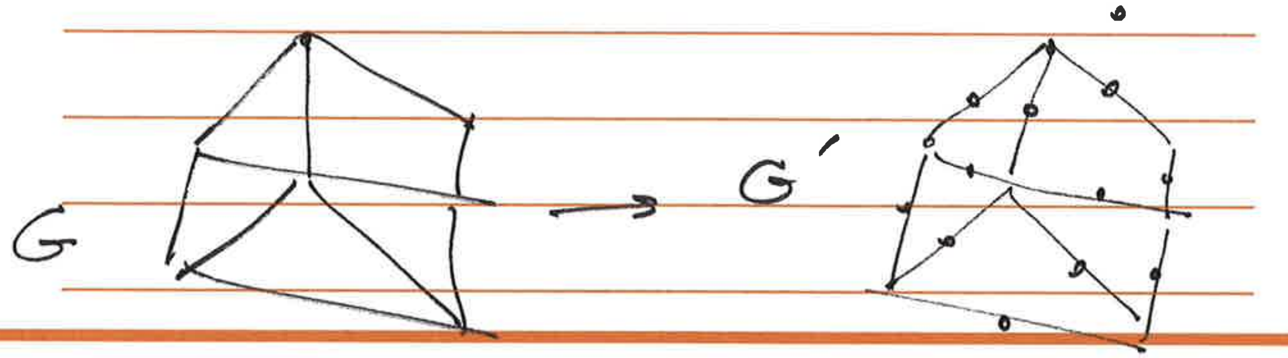
2: In the *Min-Cost Fast Path* problem, we are given a directed graph $G=(V,E)$ along with positive integer times t_e and positive costs c_e on each edge. The goal is to determine if there is a path P from s to t such that the total time on the path is at most T and the total cost is at most C (both T and C are parameters to the problem). Prove that this problem is **NP**-complete.

3: We saw in lecture that finding a Hamiltonian Cycle in a graph is **NP**-complete. Show that finding a Hamiltonian Path -- a path that visits each vertex exactly once, and isn't required to return to its starting point -- is also **NP**-complete.

1- Show that the problem of finding a HC in a bipartite graph is in NP.

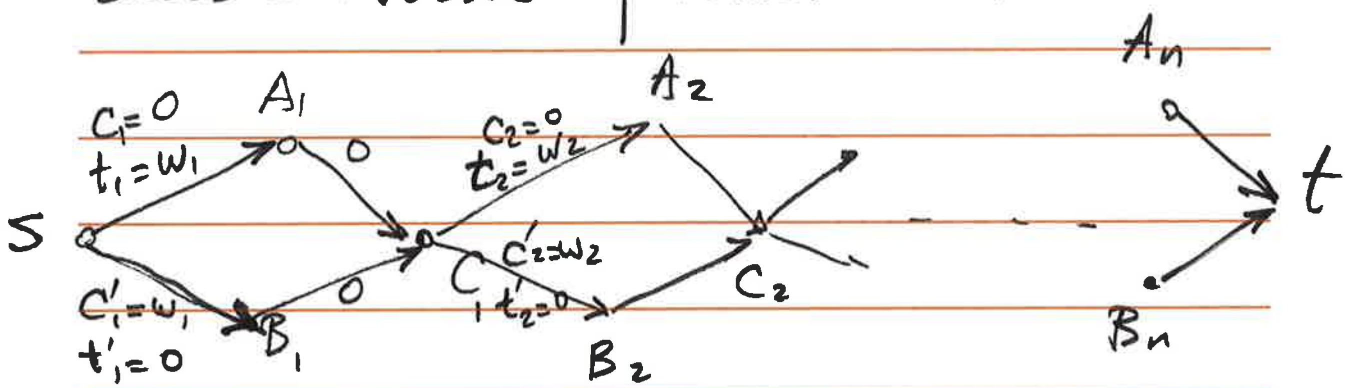
2- Choose HC

3- Reduce HC to special HC



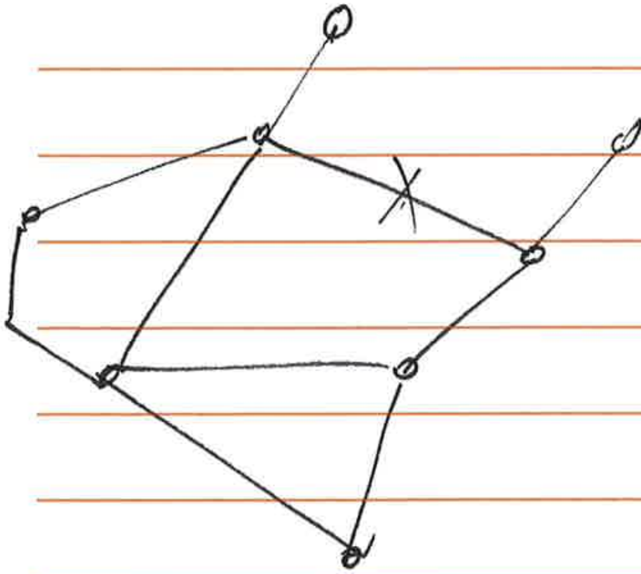
Given a set of n items each with weight w_i, is there a subset S of these items with total weight at least X and at most W?

Decision version of subsetSUM.

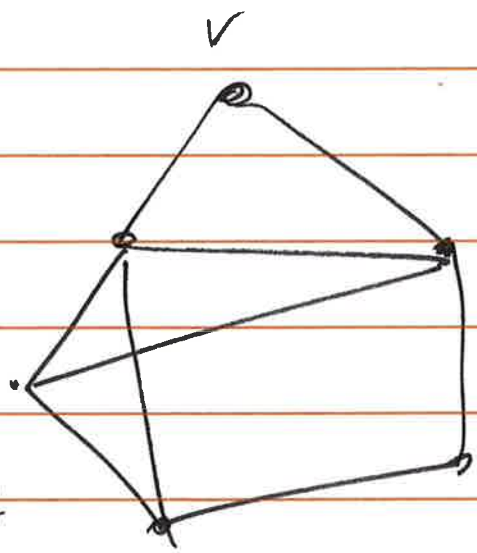


Is there an s-t path w/ total time $\leq W$

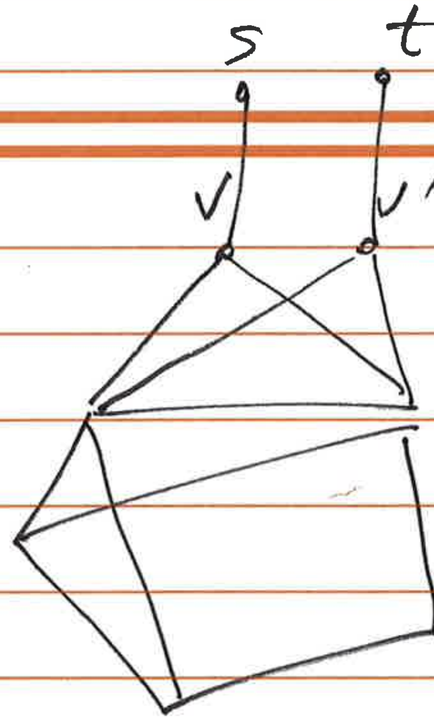
total cost $\leq \sum w_i - X$



G



G



G'

1: BDHC is in **NP**: The same proof that Hamiltonian Cycle itself is in **NP** suffices here.

Reduction: Suppose we have want to decide if a directed graph *G* has a Hamiltonian Cycle. Our goal is to create a bipartite graph *G'* such that *G'* has a Hamiltonian Cycle if and only if *G* does. To form *G'*, start with a copy of *G* and add another *n* vertices: for each vertex *u*, add a vertex *u'* and an edge *(u,u')*. Replace each edge *(u,v)* in the original graph with an edge *(u', v)*. The resulting graph is bipartite and has a Hamiltonian Cycle if and only if the original does.

2: In **NP:** Given a path, we can count its total cost and total length, and confirm that the path is valid and meets the constraints given.

Reduction: Suppose we wanted to solve Subset Sum and had a solver for this available. We can create a start vertex, plus three more for each number in the subset sum instance, called Ai, Bi, and Ci. Visiting Ai will indicate that yes, S[i] is in the desired subset, and visiting Bi will indicate that it isn't. Both will get to Ci, which will serve as the start vertex for S[i+1], or the end vertex in the case of the last number.

Add an edge from start vertex for i-1 to Ai, with zero cost and S[i] time. And an edge from start vertex for i-1 to Bi, with S[i] cost and 0 time. Both Ai and Bi connect to Ci with zero cost and zero time.

Now we ask if there's a path from start to the end with total time at most our target number from subset sum, and with total cost at most (sum of elements in subset sum minus target value). Note that one exists if and only if the subset sum does -- we'll collect the time values when we visit Ai -- the paths from a start to Ai and then to the next start add that much total time. The rest add the remaining cost.

3: 1- You duplicate a node in G (say v's duplicate is v') with all its connections and add extra nodes s and t. You connect s to v and connect t to v'. You then ask the blackbox if there is a Hamiltonian path in this graph. If there is one, obviously the original graph must have had a Hamiltonian Cycle

2- You add extra nodes s and t and connect s to one end of edge e and connect t to the other end of the same edge. You then ask the blackbox if there is a Hamiltonian path in this graph. If there is one, obviously the original graph must have had a Hamiltonian Cycle. You then repeat this for every edge. In other words you have to call the blackbox m times (once for each edge) if any of them return a Hamiltonian path, then there is a Hamiltonian cycle in the original graph.

The second reduction is interesting because it shows that you can call the black box more than once to solve the problem (polynomial number of times). In all our other reductions we have been calling the black box only once.