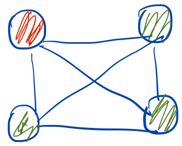
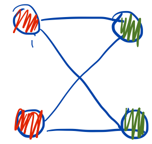




CSCI 3104: Algorithms
Spring 2022
Recitation #15 - Complexity II

SOLUTIONS

Problem 1

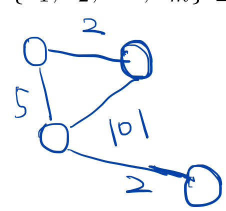
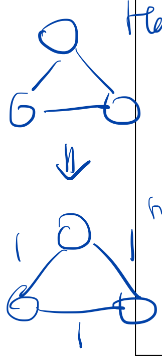
need to be able to get from any edge vert to any other edge vert in $d-1$ steps exactly... create either 1 or 2

For each of the following problems, argue whether it is a) in P or b) NP-complete.

a. Bipartite Determination. Given a graph $G = (V, E)$, is G bipartite (that is, can we partition $V = V_1 \cup V_2$ such that V_1, V_2 are disjoint and for all $u, v \in V_i$, $(u, v \notin E$?))

> This is in P. We perform a BFS, labelling nodes "1" or "0" on the odd or even iterations of our algorithm, respectively. If we ever examine an already-examined node of the wrong parity, terminate and return False, otherwise return True. BFS takes polynomial time.

b. Heavy Cycle Detection. Given a (nonnegatively) weighted graph $G = (V, E, w)$ and number $k \geq 0$, is there a simple cycle of weight at least k?

> This is NP-complete.
>
> First, we claim this is in NP. Given a cycle as an ordering of vertices $C = \{v_1, v_2, \ldots, v_m\} \subseteq V$, we check that:
>
> check vertices in G
> - $(v_i, v_{i+1})\ inV$ for all $1 \leq i \leq m-1$ and $(v_m, v_1)\ inV$ $O(m)$
>
> check #weights
> - the total weight of edges taken is at least k $O(m)$
>
> check "simple"
> - no vertices appear twice in C $O(m^2)$
>
> 
>
>
> the first two steps takes $O(n)$ time, and the final one takes at most $O(n^2)$ time, so all checks can be completed in polynomial time.
>
> 
>
>
> Ham'
>
> Seconds, we show that this is NP-complete via reduction from Hamiltonian Cycle. Given a graph $G = (V, E)$, run Heavy Cycle Detection$(G = (V, E, 1), k = n)$, where all edges have weight 1 and we ask if there is a simple cycle of weight at least n.
>
> heavy cycle
>
> No simple cycle can have weight greater than n (it would have to take more edges than there are vertices and thus visit some vertex twice), and a simple cycle of weight n visits exactly all vertices once.
>
> Thus, a Hamiltonian cycle satisfies the Heavy Cycle Detection problem, and a heavy cycle detected corresponds to a Hamiltonian cycle on the original graph.

c. Unit-Weight Knapsack. Given a capacity $C \in \mathbb{R}$ and set S of objects, each of which has weight 1 and value v_i, can we choose a subset of items with total weight at most C and total value $\geq k$?

This is in P.

CAUTION: Unconstrained Knapsack is NP-complete. The dynamic programming formulation is pseudopolynomial, meaning it is polynomial in the size of inputs rather than the number of inputs.

The greedy-by-largest-value algorithm produces the most valuable collection, which can be computed in $O(n \log n)$ time for the sort.