

CS 624: Analysis of Algorithms

Assignment 6

Solutions

1. A directed graph $G = (V, E)$ is said to be semi-connected if, for all pairs of vertices $u, v \in V$ we have $u \rightsquigarrow v$ or $v \rightsquigarrow u$ path (or both).

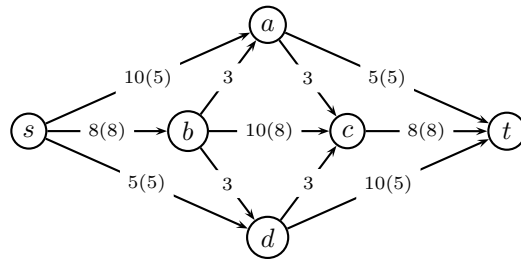
Give an efficient algorithm to determine whether or not any directed acyclic graph (DAG) $G = (V, E)$ is semi-connected.

Answer: If the graph is a DAG, then obviously there are no two vertices $u, v \in V$ such that $u \rightsquigarrow v$ AND $v \rightsquigarrow u$. However, there can be a case where for all pairs of vertices $u, v \in V$ we have $u \rightsquigarrow v$ or $v \rightsquigarrow u$. This means that every pair of vertices has an order (remember that in general, there may be vertices in a DAG that are not ordered with respect to one another, so that there is no path between them in any direction). This means that for such a DAG there can be only one possible topological sort (think why). So, a possible algorithm would be to topologically sort the DAG and for every two consecutive vertices in the topological sort, check whether there is a path v_i to v_{i+1} . By induction, this is a sufficient condition to show that there is a path between every two vertices.

2. Professor Bacon wants to rewrite the strongly connected components algorithm and use the original graph (rather than the transpose) in the second DFS run and scan the vertices in *increasing* finish time rather than decreasing. Does this modified algorithm always produce the correct results?

Answer: No. See example in class. Starting from f (whose finish time is 4), will discover g but also h , which belongs to a different component.

3. **Max-Flow-Min-Cut:** Given the following flow network on which an s-t flow has been computed. The capacity of each edge appears as a label on the edge, and the numbers in parentheses give the amount of flow sent on each edge. (Edges without parentheses – specifically, the four edges of capacity 3 – have no flow being sent on them.)



- (a) What is the value of this flow? Is this a maximum (s,t) flow in this graph?

The value of the flow is 18, as can be seen by the sum of the flow on the edges leaving s or entering t .

- (b) Find a minimum s-t cut in the flow network and also say what its capacity is.

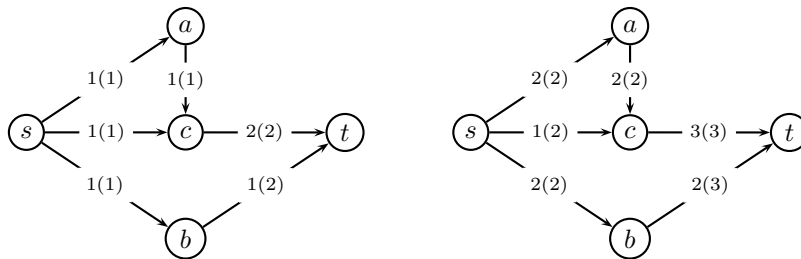
The minimum cut is $\{s, a\}$ and its capacity is 21. It is also the value of the maximum flow. It can be obtained by the following residual network below. The augmenting paths (in no particular order) are:

$s \rightarrow a \rightarrow t$ (cap. 5)
 $s \rightarrow b \rightarrow c \rightarrow t$ (cap. 5)
 $s \rightarrow a \rightarrow c \rightarrow t$ (cap. 3)
 $s \rightarrow d \rightarrow t$ (cap. 5)
 $s \rightarrow b \rightarrow d \rightarrow t$ (cap. 3).

- (c) Decide whether the following statement is true or false. If it is true, give a short explanation. If it is false, give a counter example:

Given an arbitrary flow network, with a source s , a sink t , and a positive integer capacity c_e on every edge e ; and let (A, B) be a minimum $s-t$ cut with respect to these capacities $\{c_e | e \in E\}$. Now suppose we add 1 to every capacity; then (A, B) is still a minimum $s-t$ cut with respect to these new capacities $\{1 + c_e | e \in E\}$.

The statement is false. See for example:



On the left the max-flow (and min-cut) are of value 3, and the cut is $\{s\}$ vs. $\{a, b, c, t\}$. On the right, after adding 1 to every capacity, the max flow and min-cut are 5, and the cut is now $\{s, c\}$ vs. $\{a, b, t\}$.

4. Exercise 1.1 (page 3 in the NP notes).

Answer: The first graph has a Hamiltonian cycle, for example: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow H \rightarrow G \rightarrow F \rightarrow E \rightarrow A$.

The second graph has a Hamiltonian cycle, for example: $A \rightarrow B \rightarrow C \rightarrow I \rightarrow H \rightarrow G \rightarrow A$.

The third graph doesn't have a cycle. For this we can use the lemma hinted in the question: If G is a graph that has a Hamiltonian cycle C , then every vertex of G is an endpoint of exactly 2 edges in C . Since C is a simple cycle (which it has to be, since it goes through every vertex once, see BST lecture notes), then it is of the form: $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_n \rightarrow v_1$, so that every vertex has one edge coming in and one going out.

So, in the third graph, since A, C, G and I are of degree 2, both their edges must be part of a Hamiltonian cycle if there is one. Vertex B is connected to A and C on both sides, so it must be between them in the cycle. Same for F being between C and I , H being between I and G , and D being between A and G . So this forces us to have this following order: $A \rightarrow B \rightarrow C \rightarrow F \rightarrow I \rightarrow H \rightarrow G \rightarrow D \rightarrow A$. There is no way to get to E without going through a vertex more than once. Even if we start at E or get to it somehow, we can't escape going through one of the corner vertices, which will leave us stuck.

5. Exercise 2.1 (page 8)

Answer: For a DNF expression to be satisfiable, it's enough that one of its clause is satisfiable. For a clause to be satisfiable, we need all the literals in it to have a True value. So a polynomial time algorithm to check the satisfiability of a DNF expression is as follows: For every clause, see if the clause doesn't contain a variable and its negation. If not, you can assign a True

value to all the literals in this clause and the answer to the whole question is yes. Stop and return yes.

Otherwise, move to the next clause and repeat the process. Only if all the clauses cannot be satisfied (That is, they contain a variable and its negation), return False.

This is clearly linear in the size of the expression in the worst-case, so DNF-SAT is in P.

6. Exercise 3.6 (page 15)

Answer:

- If $V1$ is a vertex cover of G , then every edge touches at least one vertex in $V1$ by definition. If we look at $V/V1$ (all the vertices not in the VC), then if $V1$ is a vertex cover, no two vertices in $V/V1$ can share an edge, because for every edge, at least one end of it touches a vertex in $V1$. Hence, $V/V1$ is an independent set.

Opposite direction: If $V/V1$ is an independent set, then no two vertices in the set share an edge. Therefore, every edge in the graph must touch at least one vertex from the remainder of the vertices, which are $V1$. This makes $V1$ a vertex cover by definition.

- If $V2$ is an independent set in G , then for each $u, v \in V2$ there is no edge (u,v) . Then, in G^c , by definition, for each $u, v \in V2$ there is an edge (u,v) , so $V2$ is a clique in G^c . The opposite direction is exactly the same.

7. Problem name: HITTING SET

Instance: A collection C of subsets of a set S together with a positive integer K .

Question: Does S contain a hitting set for C of size K or less – that is, a subset $S' \subseteq S$ with $|S'| \leq K$ and such that S' contains at least one element of each set $c \in C$?

Prove that HITTING SET is NP-complete.

To do this you need to do two things:

- (a) Prove that HITTING SET is in NP. This should be extremely easy.

Answer: Given a subset $S' \subseteq S$ and a set C , it is very easy to see whether S' is a hitting set. We check each set $c \in C$ if it contains at least one member in S' . This can be done in time linear in the total size of each subset (or even constant if we use hashing). So, given a solution – it is easy to see in polynomial time whether it is valid or not.

- (b) Prove that some problem that is already known to be NP-complete polynomially reduces to HITTING SET.

Hint: In this case, I suggest you prove that $VC \leq_P$ HITTING SET.

Answer: We start with an instance of VC, which is a graph $G = (V, E)$. S is the set of vertices, and the collection C is all the pairs of neighbors in the graph. In other words, we define a set for every $(u, v) \in E$, which contains vertices u and v . The number of sets in C is the number of edges. It is easy to see that this construction is done in polynomial time. As a matter of fact, it is linear in the size of the original graph. We only have to copy the edge information into a set of pairs of vertices.

Now we show that G has a VC of size K iff the set S has a hitting set of size K for the set of edges C :

\Rightarrow if G has a VC of size K , then we have K vertices that touch every edge. We can choose these vertices to be our hitting set for C . Since every edge in G is touched by at least one vertex from the VC, then every set $c \in C$ is hit by at least one vertex from the vertex cover.

\Leftarrow If S has a hitting set of size K , it means that we have a subset S' of K elements (vertices) such that each set $c \in C$ (the sets of edges) has at least one member in the subset S' . We can use these elements as our set cover, since the set of vertices touch every edge in the graph.

8. Dominating set:

- (a) Show that the dominating set is in NP

Answer: This is very easy. Given a set of vertices, we can easily go over the adjacency list of the vertices not in D and see that each one of these vertices has at least one member in D . This can be done in linear time with respect to the number of edges in the graph.

- (b) Show that it is NP complete by constructing a reduction from Vertex Cover (these are indeed similar problems).

Answer: We start with an instance of VC, which is a graph $G = (V, E)$ and build a graph $H = (V', E')$ as follows: The set of vertices in H is the same as in G , plus a vertex for every edge in G . Overall, H has $|V| + |E|$ vertices. There is an edge in H between every pair of vertices representing original vertices in G (in other words, these vertices constitute a clique in H) and an edge between every vertex in H representing an edge in G and its original vertices. This is done in polynomial time.

Now we show that G has a VC of size K iff H has a dominating set of size K . As a matter of fact, these are the same vertices in both graphs.

\Rightarrow if G has a VC S of size K , then the K vertices touch every edge. We can choose these vertices to be our dominating set for H . They touch every edge in G , so for every vertex in H representing an edge in G there is at least one neighbor in S . Since all the other vertices are a clique in H , then every other vertex not in S touches the vertices in S .

\Leftarrow If H has a dominating set of size K , it must have a dominating set of size K made only of original vertices. We can always replace an edge vertex e by one of its endpoints, since this endpoint dominates both the edge vertex and the other neighbor (since all original vertices in G are inter-connected in H). Since it is a dominating set, it means that every edge vertex is dominated by at least one member of the set, so in the original graph each edge is touched by at least one member of the set.