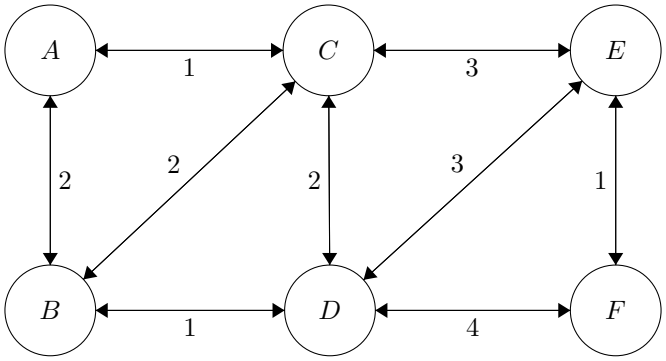
CS 61B Fall 2020

Graphs

Discussion 11: March 11, 2019

1 Minimum Spanning Trees

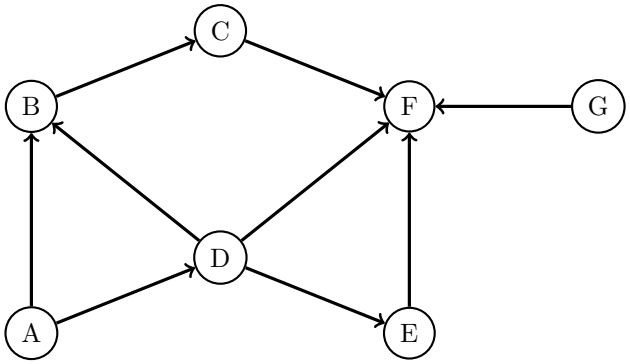


(a) Perform Prim's algorithm to find the minimum spanning tree. Pick A as the initial node. Whenever there is more than one option with the same cost, process them in alphabetical order.

(b) Use Kruskal's algorithm to find a minimum spanning tree. When deciding between equiweighted edges, alphabetically sort the edges, and then pick in lexicographic order.

For instance, edges are always written as AB or AC, never BA or CA. If deciding between AB and AC, pick AB first.

2 Topological Sorting



(a) Give a valid topological sort of the graph above. For your reference, some orderings of the graph are provided below.

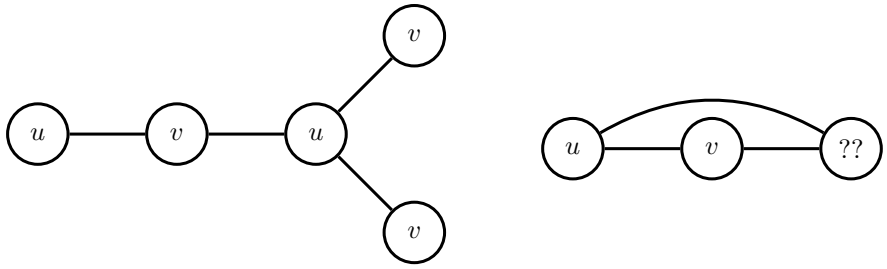
```
DFS preorder: ABCFDE (G)
DFS postorder: FCBEDA (G)
BFS: ABDCEF (G)
```

(b) There are two requirements that a graph must satisfy in order for there to be a valid topological sorting of the graph. What are they?

(c) *Extra:* Why does using a reverse post-order DFS to compute the topological sort work?

3 Graph Algorithm Design

(a) An undirected graph is said to be bipartite if all of its vertices can be divided into two disjoint sets U and V such that every edge connects an item in U to an item in V. For example below, the graph on the left is bipartite, whereas on the graph on the right is not. Provide an algorithm which determines whether or not a graph is bipartite. What is the runtime of your algorithm?



(b) Consider the following implementation of DFS, which contains a crucial error:

```
create the fringe, which is an empty Stack
push the start vertex onto the fringe and mark it
while the fringe is not empty:
    pop a vertex off the fringe and visit it
    for each neighbor of the vertex:
        if neighbor not marked:
            push neighbor onto the fringe
            mark neighbor
```

Give an example of a graph where this algorithm may not traverse in DFS order.

(c) *Extra:* Provide an algorithm that finds the shortest cycle (in terms of the number of edges used) in a directed graph in $O(EV)$ time and $O(E)$ space, assuming $E > V$.