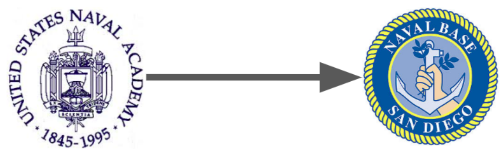
1. The Superintendent asks you and your OR classmate to give him directions from the Yard to Naval Base San Diego, the principal homeport of the Pacific Fleet. Of course, he wants to find the path that minimizes his total travel time, but he has a few more requirements: he only wants to pass through cities that have Navy Lodges, and he never wants to drive more than eight hours at a stretch between cities.



Your classmate defines the following sets and parameters:

$\mathcal{C}$:= the set of cities in the United States that have Navy Lodges;

$\mathcal{A}$:= the set of routes (i, j) from city i to city j, for i, j in $\mathcal{C}$, such that the drive time from i to j is no more than 8 hours;

$t_{i,j}$:= the drive time from city i to city j, for all (i, j) in $\mathcal{A}$;
a := the element of $\mathcal{C}$ representing Annapolis;
s := the element of $\mathcal{C}$ representing San Diego.

(a) (10 points) Picking up where your partner left off, write a mathematical program in **abstract** form whose solution solves the Superintendent's routing problem. Clearly define any additional notation that you use.

(b) (4 points) Additional research reveals that due to traffic, traveling through many of the cities $i \in \mathcal{C}$ incurs delays, and your classmate defines the parameters,

$$d_i \quad := \quad \text{the extra time required if driving through city } i, \text{ for all } i \in C.$$

Adjust your model to incorporate city delays. Again, clearly define any additional notation that you use.

(c) After you brief the Superintendent on the optimal route, he makes a few requests due to the locations of various friends and relatives. Use the variables you have already defined to add constraints to the model that will enforce the Superintendent's latest requests.

i. (2 points) He definitely wants to visit city 10, but not city 11.

ii. (2 points) He wishes to visit at least one of cities 3, 4 and 5.

iii. (2 points) If he visits cities 3 and 4, he definitely does not want to visit city 5.

7. Given the graph G in Figure 7 below, execute Prim's algorithm to calculate a minimum spanning tree starting at vertex A. Show your steps in the table below.

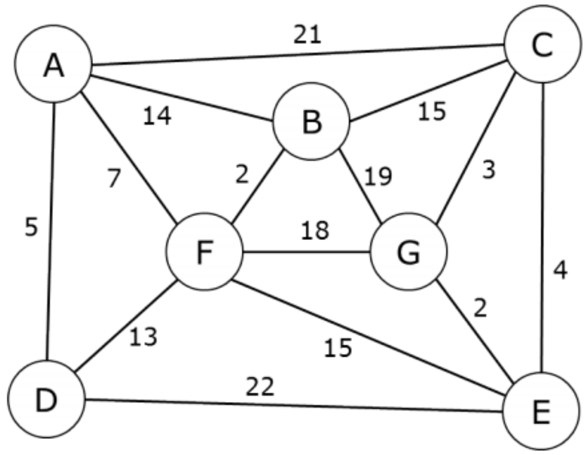


Figure 1: Graph G

(a) (15 points) For each iteration, provide all nodes currently in S, all edges in the cut-set $C_{S,\bar{S}}$, and the edge that should be added to the minimum spanning tree T. Break any ties alphabetically.

Iteration	Set S	Cut-set $C_{S,\bar{S}}$	Add edge to T
1			
2			
3			
4			
5			
6			

(b) (5 points) Indicate the minimum cost spanning tree produced by the algorithm in graph G above, and provide its cost.

8. Given the graph H in Figure 8 below, execute the Bellman-Ford algorithm to determine a shortest path from node 1 to node 6.

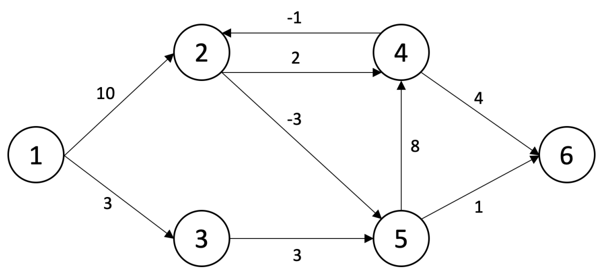


Figure 2: Graph H

To implement the Bellman-Ford algorithm, use the following ordering of edges:

$$(1,2),\ (1,3),\ (2,4),\ (2,5),\ (3,5),\ (4,2),\ (4,6),\ (5,4),\ (5,6)$$

(a) (10 points) Initialize each node with an appropriate distance label. Use the table below to track changes in distance labels, d, and predecessors, $pred$. Break any ties by choosing the minimum node label.

Vertex	1	2	3	4	5	6
d						
$pred$						

(b) (5 points) Provide the final vector of distance labels d and the final vector of predecessor labels $pred$.

$d = \{ \quad , \quad , \quad , \quad , \quad , \quad \}$

$pred = \{ \quad , \quad , \quad , \quad , \quad , \quad \}$

(c) (5 points) Provide a list of the shortest paths provided by the algorithm.