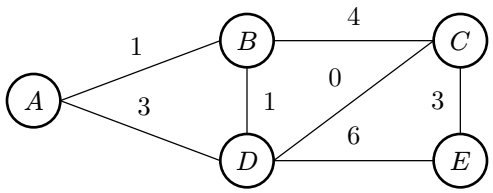
CS 61B Spring 2025

Shortest Paths, MSTs

Exam-Level 09: March 31, 2025

1 Dijkstra's, A*



(a) Run Dijkstra's Algorithm on the graph above starting from vertex A. Break ties alphabetically.

i. Fill in how the priority values change below. When you remove a node from the priority queue, mark it with a check, and leave it blank for the subsequent rows.

ii. Sketch the resulting shortest paths tree in the end.

Node	A	B	C	D	E
Start	0	∞	∞	∞	∞
Iter 1	✓				
Iter 2					
Iter 3					
Iter 4					
Iter 5					

(b) We have the following heuristic values. Run A$*$, starting from A and with E as a goal. In the table below, fill in the distance to A, followed by the priority value of each node, separated by a comma.

u	A	B	C	D	E
$h(u, E)$	8	6	5	2	0

Node	A	B	C	D	E
Start	"0, 8"	∞	∞	∞	∞
Iter 1	✓				
Iter 2					
Iter 3					
Iter 4					
Iter 5					

(c) Is the heuristic admissible (it never overestimates the true distance to the goal for each vertex)? Is the heuristic consistent (does taking roundabouts through direct neighbors gives no better distance than the heuristic)?

2 Conceptual Shortest Paths

Answer the following questions regarding shortest path algorithms for a **weighted, undirected graph**. If the statement is true, provide an explanation. If the statement is false, provide a counterexample.

(a) (T/F) If all edge weights are equal and positive, the breadth-first search starting from node A will return the shortest path from a node A to a target node B.

(b) (T/F) If all edges have distinct weights, the shortest path between any two vertices is unique.

(c) (T/F) **Adding** a constant positive integer k to all edge weights will not affect any shortest path between two vertices.

(d) (T/F) **Multiplying** a constant positive integer k to all edge weights will not affect any shortest path between two vertices.

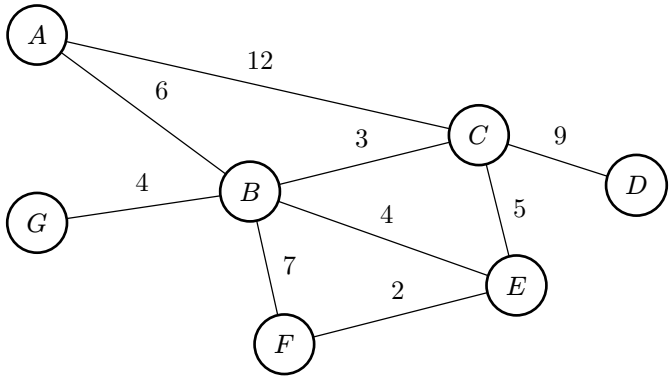
3 Shortest Paths Algorithm Design

Two cities, Chicago and Berkeley, are located in the United States. The railroad system connecting them can be modeled as a **weighted directed graph**, with V vertices, E edges, and weights representing the length of the railway. Ethan wishes to take a railway from Chicago to Berkeley, and needs to determine the shortest railway distance between them.

Define the set C to be all cities in Chicago, and B to be all cities in Berkeley. There can be cities that belong to neither region along the way. The shortest distance between the two cities is the shortest distance between any city c_C in Chicago and c_B in Berkeley.

Describe an algorithm that computes the minimum railway distance from Chicago to Berkeley, in $O((V + E) \log V)$ time. You are able to utilize all graph algorithms you learned in class. *Hint: Consider modifying the graph so that running a graph algorithm yields an equivalent answer to solving the original problem.*

4 Introduction to MSTs



(a) For the graph above, list the edges in the order they're added to the MST by Kruskal's and Prim's algorithm. Assume Prim's algorithm starts at vertex A. Assume ties are broken in alphabetical order. Denote each edge as a pair of vertices (e.g. AB is the edge from A to B).

Prim's algorithm order:

Kruskal's algorithm order:

(b) True/False: Adding 1 to the smallest edge of a graph G with unique edge weights must change the total weight of its MST.

(c) True/False: If all the weights in an MST are unique, there is only one possible MST.

(d) True/False: The shortest path from vertex u to vertex v in a graph G is the same as the shortest path from u to v using only edges in T, where T is the MST of G.