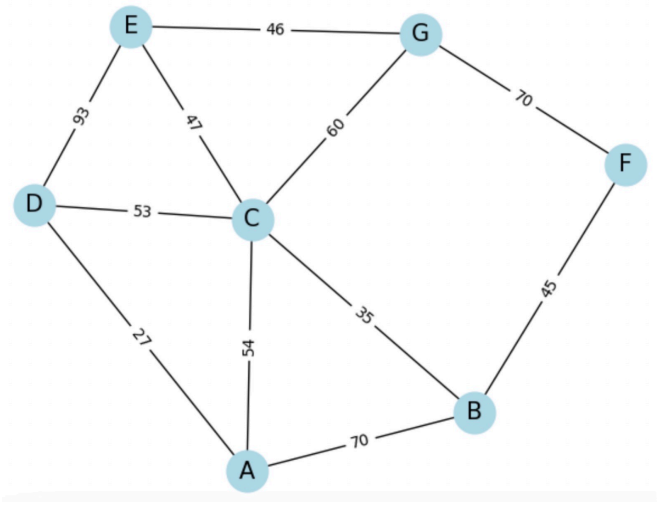
Hilde Younce



Node-City	Heuristics (H(n))
A - Washington DC	97
B - Culpeper, VA	71
C - Fredericksburg, VA	46
D - Waldorf, MD	80
E - Tappahannock, VA	41
F - Charlotesville, VA	66
G - Richmond	0

1) Djisktra algorithm : A → G

Node:	Distance:	Parent:
A	0	
B	∞	
C	∞	
D	∞	
E	∞	
F	∞	
G	∞	

visited = [ ]

unvisited = [A, B, C, D, E, F, G]

current = A

Node:	Distance:	Parent:
A	0	
B	70	A
C	54	A
D	27	A
E	∞	
F	∞	
G	∞	

visited = [A]

unvisited = [B, C, D, E, F, G]

⇒ current = D

Node:	Distance:	Parent:
A	0	
B	70	A
C	54	A
D	27	A
E	27 + 93 = 120	D
F	∞	
G	∞	

visited = [A, D]

unvisited = [B, C, E, F, G]

⇒ CURRENT = C

Node:	Distance:	Parent:
A	0	
B	70	A
C	54	A
D	27	A
E	54+47 = 101	C
F	∞	
G	54 + 60 = 114	C

visited = [A, C, D]

unvisited = [B, E, F, G]

⇒ CURRENT = B

Node:	Distance:	Parent:
A	0	
B	70	A
C	54	A
D	27	A
E	101	C
F	70+45 = 115	B
G	114	C

visited = [A, B, C, D]

unvisited = [E, F, G]

⇒ CURRENT = E

Node:	Distance:	Parent:
A	0	
B	70	A
C	54	A
D	27	A
E	101	C
F	115	B
G	114	C

visited = [A, B, C, D, E]

unvisited = [F, G]

⇒ current = G

Therefore, the shortest path from A → G is

A → C → G OR D.C → Fredericksburg → Richmond

2) A* Algorithm:

Node:	g(n):	h(n):	f(n):
A	0	97	
B	∞	71	
C	∞	46	
D	∞	80	
E	∞	41	
F	∞	66	
G	∞	0	

visited = [ ]

unvisited = [A, B, C, D, E, F, G]

⇒ current = A

Node:	g(n):	h(n):	f(n):
A	0	97	97
B	70	71	141
C	54	46	100
D	27	80	107
E	∞	41	
F	∞	66	
G	∞	0	

visited = [A]

unvisited = [B, C, D, E, F, G]

⇒ current = C

Node:	g(n):	h(n):	f(n):
A	0	97	97
B	70	71	141
C	54	46	100
D	27	80	107
E	54+47	41	142
F	∞	66	
G	54+60	0	114

visited = [A, C]

unvisited = [B, D, E, F, G]

⇒ current = D

Node:	g(n):	h(n):	f(n):
A	0	97	97
B	70	71	141
C	54	46	100
D	27	80	107
E	101	41	142
F	∞	66	
G	114	0	114

visited = [A, C, D]

unvisited = [B, E, F, G]

⇒ current = G

So, the shortest path from A → G is A → C → G