Campus Dash: Solution

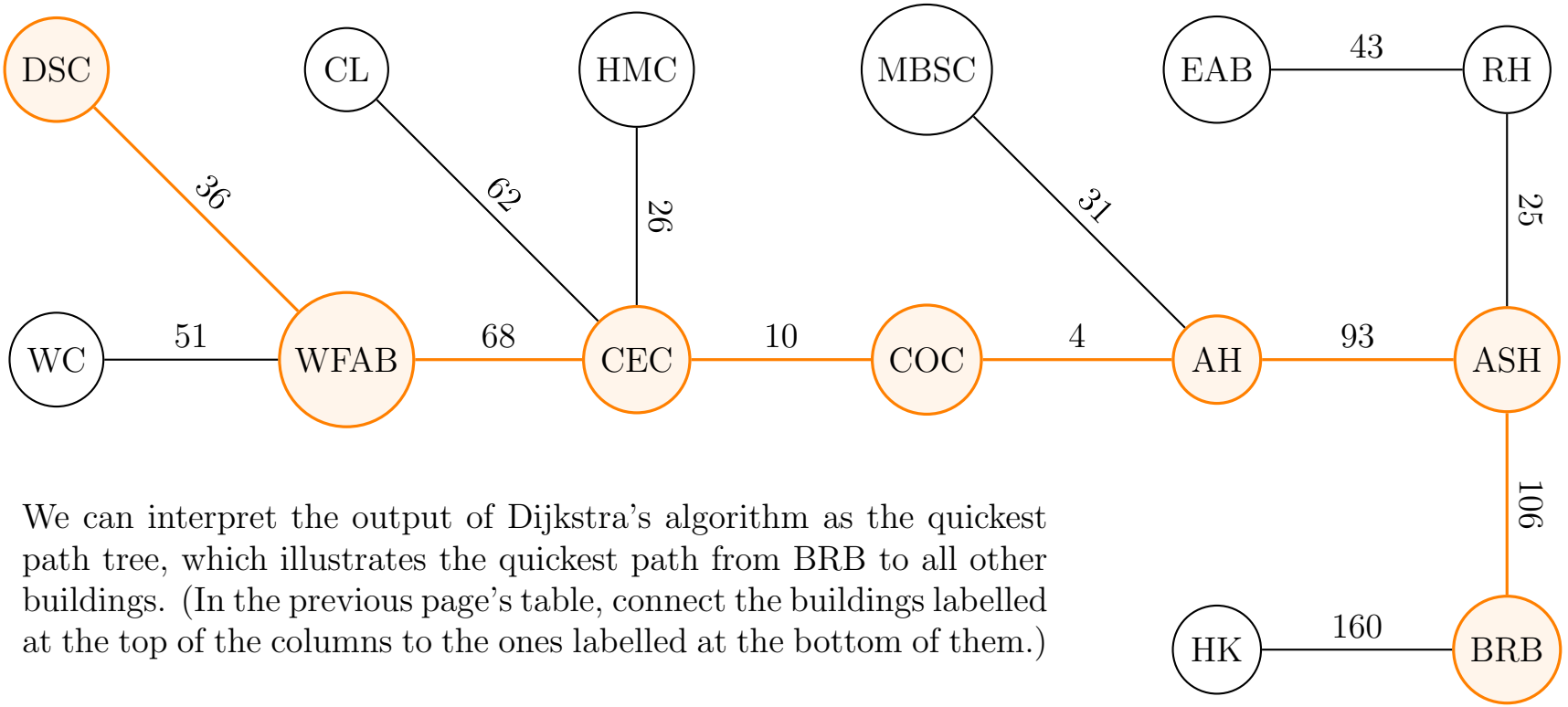
Dijkstra's algorithm builds, piece by piece, the quickest paths from BRB to all other buildings. During each pass of the algorithm, each node will have a minimum-known time from BRB, and the name of the previous node in a quickest-known path from BRB to it if applicable. The first pass of the algorithm sets all minimum-known times to ∞, except BRB's minimum-known time is set to 0. Every pass after "visits" a new node N, examining all of its unvisited neighbors U - the pass compares (the current minimum-known time from BRB to U) to (the minimum time from BRB to N plus the time from N to U): if U's current minimum-known time is larger it is replaced and the previous node is updated to N.

Pass	MBSC	COC	EAB	AH	RH	ASH	HK
2	-	-	-	-	-	106 BRB	260 BRB
3	-	-	-	199 ASH	131 ASH		
4	-	-	174 RH				
5	283 EAB	-					
6	230 AH	203 AH					

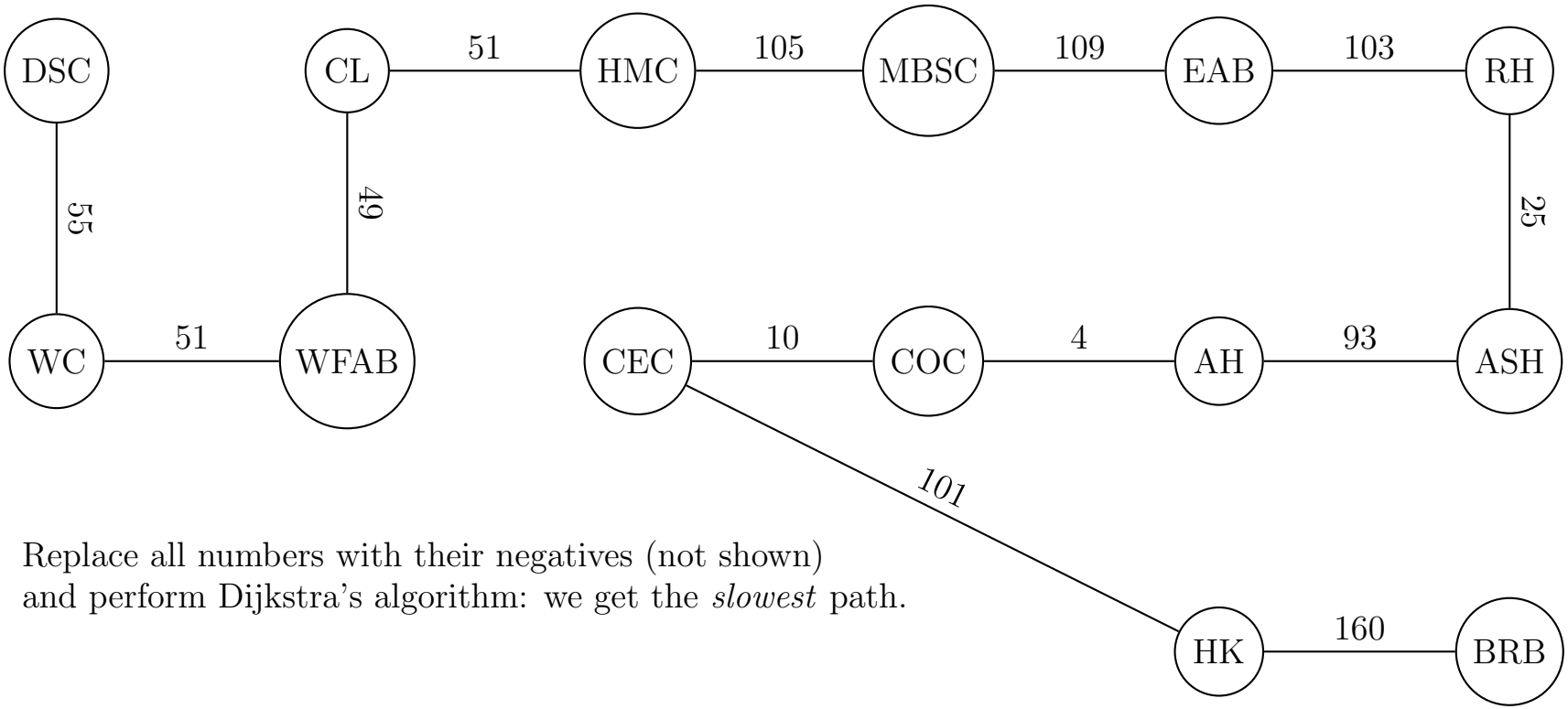
The dashes represent ∞, and whenever a minimum-time is found the rest of the column is left blank.

Pass	DSC	CL	WC	WFAB	HMC	CEC
7	-	-	-	-	-	213 COC
8	-	275 CEC	-	281 CEC	239 CEC	
9	-		-			
10	-		-			
11	-		-			
12	359 CL		-			
13	317 WFAB		332 WFAB			

The quickest path from BRB to DSC, highlighted in orange below, takes 5 minutes and 17 seconds (plus the time it takes to walk through or around the buildings themselves, which we are ignoring).



We can interpret the output of Dijkstra's algorithm as the quickest path tree, which illustrates the quickest path from BRB to all other buildings. (In the previous page's table, connect the buildings labelled at the top of the columns to the ones labelled at the bottom of them.)



Replace all numbers with their negatives (not shown)
and perform Dijkstra's algorithm: we get the *slowest* path.