Example Computation of Dijkstra's Algorithm

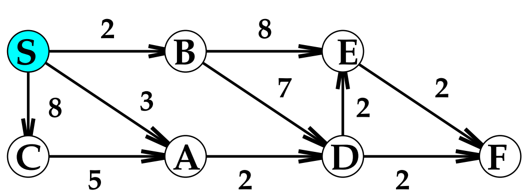
Suppose we are given the weighted directed graph G shown below. Dijkstra's algorithm solves the single sooource minpath problem for G. where S is the source vertex.

The input of the algorithm is a set of weighted edges. We let $W[i, j]$ be the weight of the edge from i to j. If there is no such edge, we let $W[i, j]$ default to infinity.

The output of Dijkstra's algorithm is an array V where $V[i]$ is the value of vertex i, the least weight of any path from S to i, and also an array if backpointers, where for any vertex i except the source vertex, back$[i]$ is the next-to-the-last vertex on the least weight path from S to i.

At each step, we show G together with the backpointers that have been computed, followed by the updatable minheap implemented as a binary tree implemented as an array, followed by a matrix with one column for each vertex. The second row of the matrix shows the array V, the third row shows the array back, and the fourth row shows the position of each partially processed vertex in the minheap. In the matrix, the columns for the vertices that have been fully processed will be shown with a colored background.

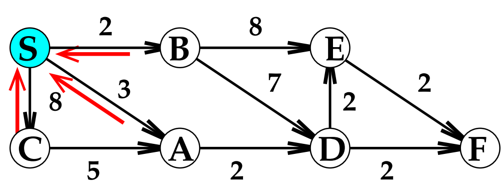
Initially, there are no backpointers and no vertex is fully processed, and the source vertex S is partially processed.



1
S

S	A	B	C	D	E	F
0	∞	∞	∞	∞	∞	∞
1						

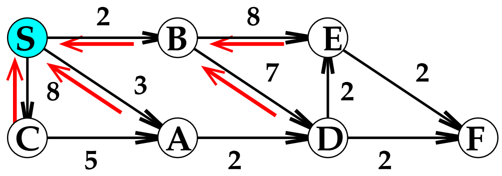
At the first phase, the minitem S is deleted from the minheap and the subsequents of S, namely A, B, and C, are inserted. Notice that heap order must be maintained, where the key of vertex i is its value $V[i]$. Backpointers are shown as red arrows in the figure; back$[A]$ = back$[B]$ = back$[C] = S$.



1	2	3
B	A	C

S	A	B	C	D	E	F
0	3	2	8	∞	∞	∞
	S	S	S			
	2	1	3			

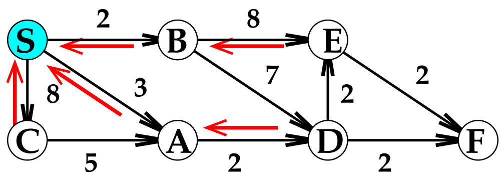
The min value vertex is always in position 1 in the meanheap. That vertex, B, is deleted, and its subsequents D and E interserted, and heap order is restored. The color background indicates that S and B are fully processed. The partially processed vertices are those in the minheap, A, C, E, and D.



1	2	3	4
A	C	E	D

S	A	B	C	D	E	F
0	3	2	8	9	10	∞
	S	S	S	B	B	
	1		2	4	3	

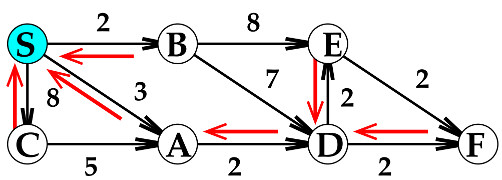
The minimum vertex A is deleted. Its subsequent, D, has acquired a new, lower value, and back[D] is now A.



1	2	3
D	C	E

S	A	B	C	D	E	F
0	3	2	8	5	10	∞
	S	S	S	A	B	
			2	4	3	

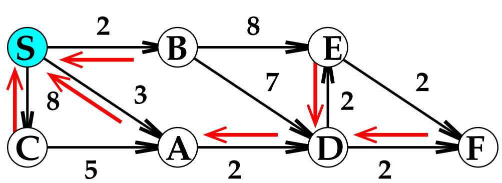
D is deleted from the minqueue, and the subsequents of D are inserted. E has a acquired a new value and a new backpointer. All vertices are now partially or fully processed.



1	2	3
E	C	F

S	A	B	C	D	E	F
0	3	2	8	5	7	7
	S	S	S	A	D	D
			2		1	3

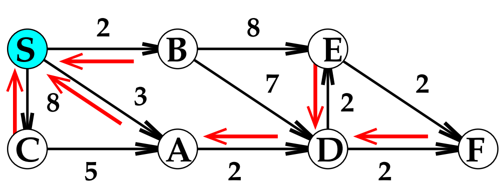
Since there is a tie for minimum value, we can delete vertices E and F simultaneously.



1
C

S	A	B	C	D	E	F
0	3	2	8	5	7	7
	S	S	S	A	D	D
			1			

The minheap is empty, and we are done.



S	A	B	C	D	E	F
0	3	2	8	5	7	7
	S	S	S	A	D	D