Prim MST: initialized the graph.
Priority queue = \([L_{\text{+inf}}, H_{\text{+inf}}, F_{\text{+inf}}, T_{\text{+inf}}, J_{\text{+inf}}, G_{\text{+inf}}, M_{\text{+inf}}, K_{\text{+inf}}]\)
Weight of red edges = = 0
Marked the source node F.
Priority queue = [F_0, L_+inf, H_+inf, T_+inf, J_+inf, G_+inf, M_+inf, K_+inf]
Weight of red edges =  = 0
Extracted F and added it to the MST-so-far. Let's adjust its adjacent vertices (H, L, G).

Priority queue = [L_{+\infty}, H_{+\infty}, T_{+\infty}, J_{+\infty}, G_{+\infty}, M_{+\infty}, K_{+\infty}]

Weight of red edges = 0

F_0

H_{+\infty}

T_{+\infty}

G_{+\infty}

J_{+\infty}

L_{+\infty}

K_{+\infty}

M_{+\infty}
Adjusting the vertices adjacent to F. Considering edge (F, H), leading to H.

Priority queue = [L_+inf, H_+inf, T_+inf, J_+inf, G_+inf, M_+inf, K_+inf]

Weight of red edges = = 0
Using edge \((F, H)\), vertex \(H\) can be reached in 3 from the MST-so-far (better than \(+\text{inf}\)).

Priority queue = \([H_3, L_+\text{inf}, T_+\text{inf}, J_+\text{inf}, G_+\text{inf}, M_+\text{inf}, K_+\text{inf}]\)

Weight of red edges = 0
So let's add edge (F, H) to the MST.

Priority queue = [H_3, L_+inf, T_+inf, J_+inf, G_+inf, M_+inf, K_+inf]

Weight of red edges = 3 = 3
Adjusting the vertices adjacent to F. Considering edge (F, L), leading to L.
Priority queue = [H_3, L_+inf, T_+inf, J_+inf, G_+inf, M_+inf, K_+inf]
Weight of red edges = 3 = 3
Using edge (F, L), vertex L can be reached in 3 from the MST-so-far (better than +inf).

Priority queue = [H_3, L_3, T_+inf, J_+inf, G_+inf, M_+inf, K_+inf]
Weight of red edges = 3 = 3
So let's add edge (F, L) to the MST.
Priority queue = [H_3, L_3, T_+inf, J_+inf, G_+inf, M_+inf, K_+inf]
Weight of red edges = 3 + 3 = 6
Adjusting the vertices adjacent to F. Considering edge (F, G), leading to G.
Priority queue = [H_3, L_3, T_+inf, J_+inf, G_+inf, M_+inf, K_+inf]
Weight of red edges = 3 + 3 = 6
Using edge (F, G), vertex G can be reached in 5 from the MST-so-far (better than +inf).

Priority queue = [H_3, L_3, G_5, T_+inf, J_+inf, M_+inf, K_+inf]

Weight of red edges = 3 + 3 = 6
So let's add edge (F, G) to the MST.
Priority queue = [H_3, L_3, G_5, T_+inf, J_+inf, M_+inf, K_+inf]
Weight of red edges = 3 + 3 + 5 = 11
Finished with the adjacents of F.
Priority queue = [H_3, L_3, G_5, T_+inf, J_+inf, M_+inf, K_+inf]
Weight of red edges = 3 + 3 + 5 = 11
Extracted H and added it to the MST-so-far. Let's adjust its adjacent vertices (K, J, T, F, M).

Priority queue = [L_3, G_5, T_+inf, J_+inf, M_+inf, K_+inf]

Weight of red edges = 3 + 3 + 5 = 11
Adjusting the vertices adjacent to H. Considering edge (H, K), leading to K.

Priority queue = [L_3, G_5, T_+inf, J_+inf, M_+inf, K_+inf]

Weight of red edges = 3 + 3 + 5 = 11
Using edge (H, K), vertex K can be reached in 7 from the MST-so-far (better than +inf).
Priority queue = [L_3, G_5, K_7, T_+inf, J_+inf, M_+inf]
Weight of red edges = 3 + 3 + 5 = 11
So let's add edge (H, K) to the MST.
Priority queue = [L_3, G_5, K_7, T_+inf, J_+inf, M_+inf]
Weight of red edges = 3 + 3 + 5 + 7 = 18
Adjusting the vertices adjacent to H. Considering edge (H, J), leading to J.

Priority queue = [L_3, G_5, K_7, T_+inf, J_+inf, M_+inf]

Weight of red edges = $3 + 3 + 5 + 7 = 18$
Using edge (H, J), vertex J can be reached in 2 from the MST-so-far (better than +inf). Priority queue = [J_2, L_3, G_5, K_7, T_+inf, M_+inf] Weight of red edges = 3 + 3 + 5 + 7 = 18
So let's add edge \((H, J)\) to the MST.
Priority queue = \([J_2, L_3, G_5, K_7, T_{+\infty}, M_{+\infty}]\)
Weight of red edges = \(3 + 3 + 5 + 7 + 2 = 20\)
Adjusting the vertices adjacent to H. Considering edge (H, T), leading to T.
Priority queue = [J_2, L_3, G_5, K_7, T_+inf, M_+inf]
Weight of red edges = 3 + 3 + 5 + 7 + 2 = 20
Using edge (H, T), vertex T can be reached in 32 from the MST-so-far (better than +inf).
Priority queue = [J_2, L_3, G_5, K_7, T_32, M_+inf]
Weight of red edges = 3 + 3 + 5 + 7 + 2 = 20
So let's add edge (H, T) to the MST.
Priority queue = [J_2, L_3, G_5, K_7, T_32, M_+\text{inf}]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 32 = 52
Adjusting the vertices adjacent to H. Considering edge (F, H), leading to F.
Priority queue = [J_2, L_3, G_5, K_7, T_32, M_+inf]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 32 = 52
Using edge (F, H), vertex F can be reached in 3 from the MST-so-far (not better than 0).
   Let's not use that edge.

Priority queue = [J_2, L_3, G_5, K_7, T_32, M_+inf]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 32 = 52
Adjusting the vertices adjacent to H. Considering edge (M, H), leading to M.

Priority queue = [J_2, L_3, G_5, K_7, T_32, M_+inf]

Weight of red edges = 3 + 3 + 5 + 7 + 2 + 32 = 52
Using edge \((M, H)\), vertex \(M\) can be reached in 6 from the MST-so-far (better than +inf).
Priority queue = \([J_2, L_3, G_5, M_6, K_7, T_{32}]\)
Weight of red edges = \(3 + 3 + 5 + 7 + 2 + 32 = 52\)
So let's add edge \((M, H)\) to the MST.

Priority queue = \([J_2, L_3, G_5, M_6, K_7, T_{32}]\)

Weight of red edges = \(3 + 3 + 5 + 7 + 2 + 32 + 6 = 58\)
Finished with the adjacents of H.
Priority queue = [J_2, L_3, G_5, M_6, K_7, T_32]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 32 + 6 = 58
Extracted J and added it to the MST-so-far. Let's adjust its adjacent vertices (L, K, H, T).

Priority queue = [L_3, G_5, M_6, K_7, T_32]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 32 + 6 = 58
Adjusting the vertices adjacent to J. Considering edge (J, L), leading to L.

Priority queue = \([L_3, G_5, M_6, K_7, T_32]\)

Weight of red edges = \(3 + 3 + 5 + 7 + 2 + 32 + 6 = 58\)
Using edge (J, L), vertex L can be reached in 5 from the MST-so-far (not better than 3). Let's not use that edge.

Priority queue = [L_3, G_5, M_6, K_7, T_32]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 32 + 6 = 58
Adjusting the vertices adjacent to J. Considering edge (J, K), leading to K.
Priority queue = [L_3, G_5, M_6, K_7, T_32]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 32 + 6 = 58
Using edge (J, K), vertex K can be reached in 23 from the MST-so-far (not better than 7).
Let's not use that edge.
Priority queue = [L_3, G_5, M_6, K_7, T_32]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 32 + 6 = 58
Adjusting the vertices adjacent to J. Considering edge (H, J), leading to H.
Priority queue = [L_3, G_5, M_6, K_7, T_32]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 32 + 6 = 58
Using edge (H, J), vertex H can be reached in 2 from the MST-so-far (not better than 0).
Let's not use that edge.
Priority queue = [L_3, G_5, M_6, K_7, T_32]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 32 + 6 = 58
Adjusting the vertices adjacent to J. Considering edge (T, J), leading to T.
Priority queue = [L_3, G_5, M_6, K_7, T_32]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 32 + 6 = 58
Using edge (T, J), vertex T can be reached in 4 from the MST-so-far (better than 32).

Priority queue = [L_3, T_4, G_5, M_6, K_7]

Weight of red edges = 3 + 3 + 5 + 7 + 2 + 32 + 6 = 58
So let's add edge (T, J) to the MST, replacing (H, T).
Priority queue = [L_3, T_4, G_5, M_6, K_7]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 6 + 4 = 30
Finished with the adjacents of J.
Priority queue = [L_3, T_4, G_5, M_6, K_7]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 6 + 4 = 30
Extracted L and added it to the MST-so-far. Let's adjust its adjacent vertices (M, J, F).
Priority queue = [T_4, G_5, M_6, K_7]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 6 + 4 = 30
Adjusting the vertices adjacent to L. Considering edge (L, M), leading to M.
Priority queue = [T_4, G_5, M_6, K_7]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 6 + 4 = 30
Using edge (L, M), vertex M can be reached in 1 from the MST-so-far (better than 6).
Priority queue = [M_1, T_4, G_5, K_7]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 6 + 4 = 30
So let's add edge (L, M) to the MST, replacing (M, H).
Priority queue = [M_1, T_4, G_5, K_7]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Adjusting the vertices adjacent to L. Considering edge (J, L), leading to J.
Priority queue = [M_1, T_4, G_5, K_7]
Weight of red edges = $3 + 3 + 5 + 7 + 2 + 4 + 1 = 25$
Using edge (J, L), vertex J can be reached in 5 from the MST-so-far (not better than 0).

Let's not use that edge.

Priority queue = \([M_1, T_4, G_5, K_7]\)

Weight of red edges = \(3 + 3 + 5 + 7 + 2 + 4 + 1 = 25\)
Adjusting the vertices adjacent to L. Considering edge (F, L), leading to F.
Priority queue = [M_1, T_4, G_5, K_7]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Using edge (F, L), vertex F can be reached in 3 from the MST-so-far (not better than 0).
Let's not use that edge.

Priority queue = [M_1, T_4, G_5, K_7]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Finished with the adjacents of L.
Priority queue = [M_1, T_4, G_5, K_7]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Extracted M and added it to the MST-so-far. Let’s adjust its adjacent vertices (T, H, L).

Priority queue = [T_4, G_5, K_7]

Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Adjusting the vertices adjacent to M. Considering edge (M, T), leading to T.

Priority queue = [T_4, G_5, K_7]

Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Using edge (M, T), vertex T can be reached in 8 from the MST-so-far (not better than 4). Let's not use that edge.

Priority queue = [T_4, G_5, K_7]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Adjusting the vertices adjacent to M. Considering edge (M, H), leading to H.
Priority queue = [T_4, G_5, K_7]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Using edge (M, H), vertex H can be reached in 6 from the MST-so-far (not better than 0). Let's not use that edge.

Priority queue = [T_4, G_5, K_7]

Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Adjusting the vertices adjacent to M. Considering edge (L, M), leading to L.

Priority queue = [T_4, G_5, K_7]

Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Using edge (L, M), vertex L can be reached in 1 from the MST-so-far (not better than 0).

Let's not use that edge.

Priority queue = [T_4, G_5, K_7]

Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Finished with the adjacents of M.
Priority queue = [T_4, G_5, K_7]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Extracted T and added it to the MST-so-far. Let's adjust its adjacent vertices (G, J, M, H).

Priority queue = [G_5, K_7]

Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Adjusting the vertices adjacent to T. Considering edge (T, G), leading to G.
Priority queue = [G_5, K_7]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Using edge (T, G), vertex G can be reached in 10 from the MST-so-far (not better than 5). Let's not use that edge.

Priority queue = [G_5, K_7]

Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Adjusting the vertices adjacent to T. Considering edge (T, J), leading to J.

Priority queue = [G_5, K_7]

Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Using edge \((T, J)\), vertex \(J\) can be reached in 4 from the MST-so-far (not better than 0).

Let's not use that edge.

Priority queue = \([G_5, K_7]\)

Weight of red edges = \(3 + 3 + 5 + 7 + 2 + 4 + 1 = 25\)
Adjusting the vertices adjacent to T. Considering edge (M, T), leading to M.
Priority queue = [G_5, K_7]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Using edge \((M, T)\), vertex \(M\) can be reached in 8 from the MST-so-far (not better than 0).

Let's not use that edge.

Priority queue = \([G_5, K_7]\)

Weight of red edges = \(3 + 3 + 5 + 7 + 2 + 4 + 1 = 25\)
Adjusting the vertices adjacent to T. Considering edge (H, T), leading to H.
Priority queue = [G_5, K_7]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Using edge (H, T), vertex H can be reached in 32 from the MST-so-far (not better than 0). Let's not use that edge.

Priority queue = [G_5, K_7]

Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Finished with the adjacents of T.
Priority queue = \([G_5, K_7]\)
Weight of red edges = \(3 + 3 + 5 + 7 + 2 + 4 + 1 = 25\)
Extracted G and added it to the MST-so-far. Let's adjust its adjacent vertices (T, F).

Priority queue = [K_7]

Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Adjusting the vertices adjacent to \( G \). Considering edge \((T, G)\), leading to \( T \).

Priority queue = \([K_7]\)

Weight of red edges = \(3 + 3 + 5 + 7 + 2 + 4 + 1 = 25\)
Using edge \((T, G)\), vertex \(T\) can be reached in 10 from the MST-so-far (not better than 0). Let’s not use that edge.

Priority queue = \([K_7]\)

Weight of red edges = \(3 + 3 + 5 + 7 + 2 + 4 + 1 = 25\)
Adjusting the vertices adjacent to G. Considering edge (F, G), leading to F.
Priority queue = [K_7]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Using edge (F, G), vertex F can be reached in 5 from the MST-so-far (not better than 0). Let's not use that edge.

Priority queue = [K_7]

Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Finished with the adjacents of G.
Priority queue = [K_7]
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Extracted K and added it to the MST-so-far. Let's adjust its adjacent vertices (H, J).

Priority queue = []

Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Adjusting the vertices adjacent to K. Considering edge (H, K), leading to H.
Priority queue = []
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Using edge (H, K), vertex H can be reached in 7 from the MST-so-far (not better than 0). Let's not use that edge.

Priority queue = []

Weight of red edges = $3 + 3 + 5 + 7 + 2 + 4 + 1 = 25$
Adjusting the vertices adjacent to K. Considering edge (J, K), leading to J.
Priority queue = []
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
Using edge (J, K), vertex J can be reached in 23 from the MST-so-far (not better than 0).
Let's not use that edge.
Priority queue = []
Weight of red edges = $3 + 3 + 5 + 7 + 2 + 4 + 1 = 25$
Finished with the adjacents of K.
Priority queue = []
Weight of red edges = 3 + 3 + 5 + 7 + 2 + 4 + 1 = 25
MST now complete.
Priority queue = []
Weight of red edges = $3 + 3 + 5 + 7 + 2 + 4 + 1 = 25$
Prim minimum spanning tree
Generated by $Id: prim.py 87 2010-11-15 23:48:22Z fms27$
(c) 2006-2010 Frank Stajano
University of Cambridge Computer Laboratory