CS300 - Data Structures
HW5

Emir Alaattin Yılmaz

Question 1. We assumed that each size of partition is equally likely and hence, has probability 1/N. With given assumptions we get the following running time equation:

$$T\ \mathrm{N}\ = \frac{1}{N}\sum_{\mathrm{j}=0}^{\mathrm{N}-1} \mathrm{T\ j}\ + cN \tag{1}$$

If we multiply both sides of Eq.1 with N we have:

$$NT\ N\ = \sum_{j=0}^{N-1} T\ j\ + cN^2 \tag{2}$$

To remove summation sign for simplification, we telescoped one step:

$$(N-1)T\ N-1\ = \sum_{j=0}^{N-2} T\ j\ + c(N-1)^2 \tag{3}$$

Subtracting Eq.2 from Eq.3 we get:

$$NT\ N\ \ -\ (N-1)T(N-1) = T(N-1) + 2cN\ -\ c \tag{4}$$

After rearrangement and drop the insignificant -c on the right, obtaining

$$NT\ N\ = NT(N-1) + 2cN \tag{5}$$

Divide Eq.5 by N we have:

$$T\ N\ = T(N-1) + 2c \tag{6}$$

Now we can telescope.

$$T\ N\ = T(N-1) + 2c \tag{7}$$

$$...$$

$$T\ 2\ = T(1) + 2c \tag{8}$$

Adding equations (7) through (9) yields

$$T\ N\ = T(1) + (N-1)2c \tag{9}$$

Since no cut-off size used, T(1) = 0, therefore

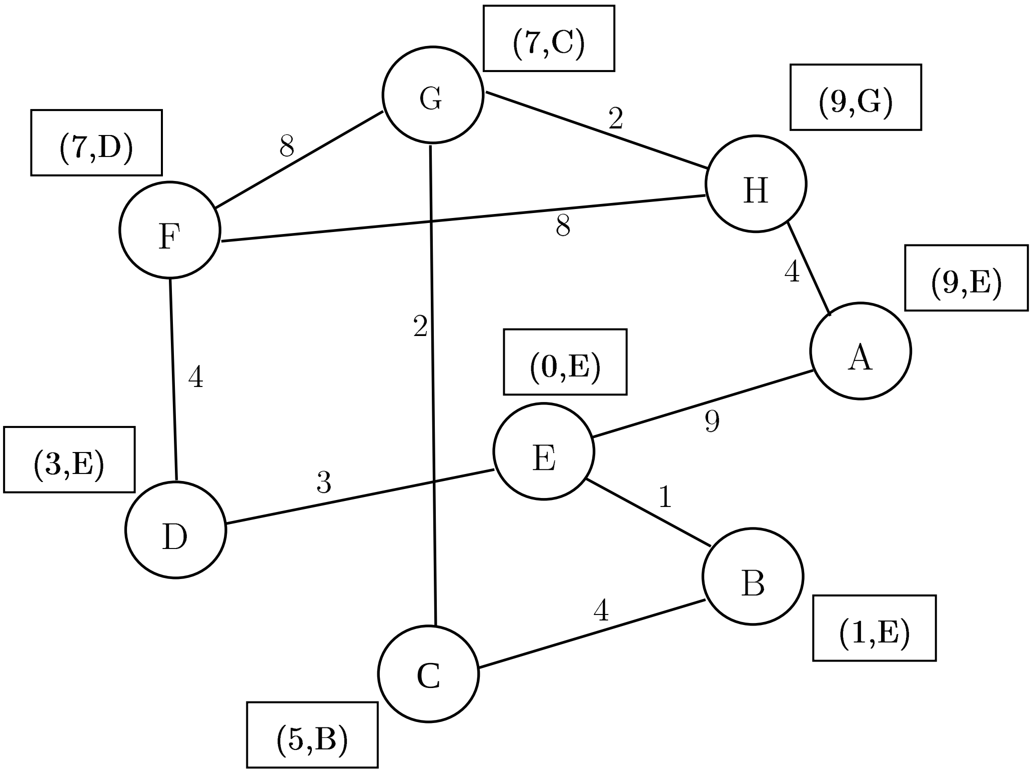
$$T\ N\ = (N-1)2c\ =\ O(N) \tag{10}$$

Question 2. Dijkstra Algorithm

KNOWN VERTICES	d_A,p_A	d_B,p_B	d_C,p_C	d_D,p_D	d_E,p_E	d_F,p_F	d_G,p_G	d_H,p_H
E	(9,E)	(1,E)	∞	(3,E)	(0,E)	∞	∞	∞
E,B	(9,E)	-	(5,B)	(3,E)	-	∞	∞	∞
E,B,D	(9,E)	-	(5,B)	-	-	(7,D)	∞	∞
E,B,D,C	(9,E)	-	-	-	-	(7,D)	∞	∞
E,B,D,C,F	(9,E)	-	-	-	-	-	(7,C)	(15,F)
E,B,D,C,F,G	(9,E)	-	-	-	-	-	-	(9,G)
E,B,D,C,F,G,H	(9,E)	-	-	-	-	-	-	-
E,B,D,C,F,G,H,A	-	-	-	-	-	-	-	-

***(d_X,p_Y) = (distance to vertex X, came from vertex Y)**

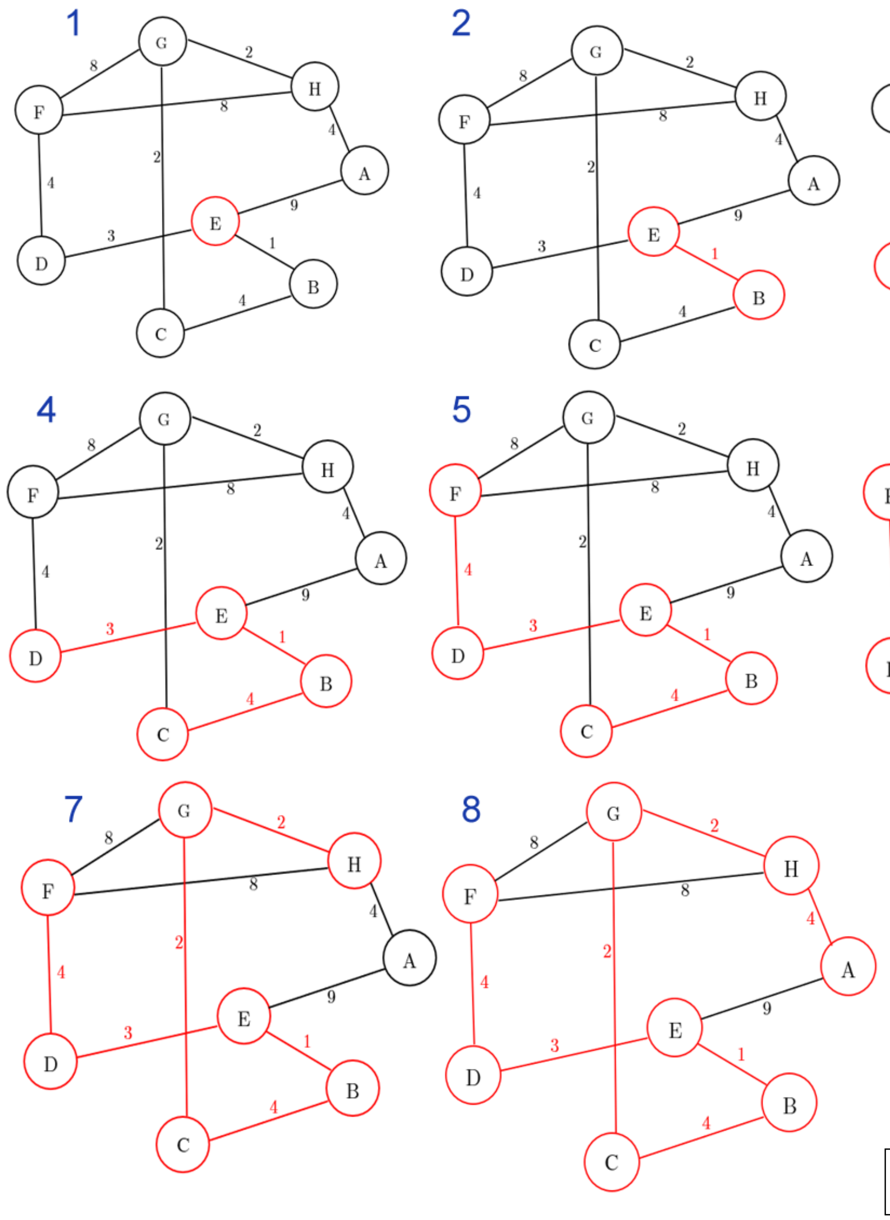
Shortest path to every vertex from E



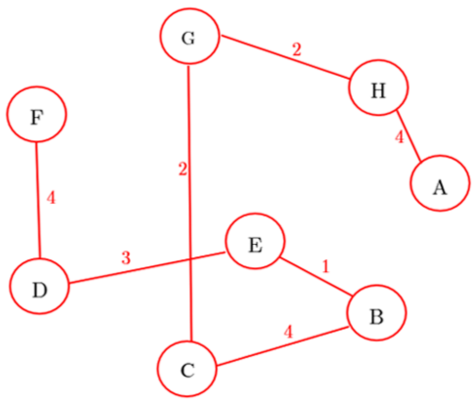
Shortest Path Table

A	B	C	D	E	F	G	H
(9,E)	(1,E)	(5,B)	(3,E)	(0,E)	(7,D)	(7,C)	(9,G)

Question 3. Prim's Algorithm



Minimum Spanning Tree

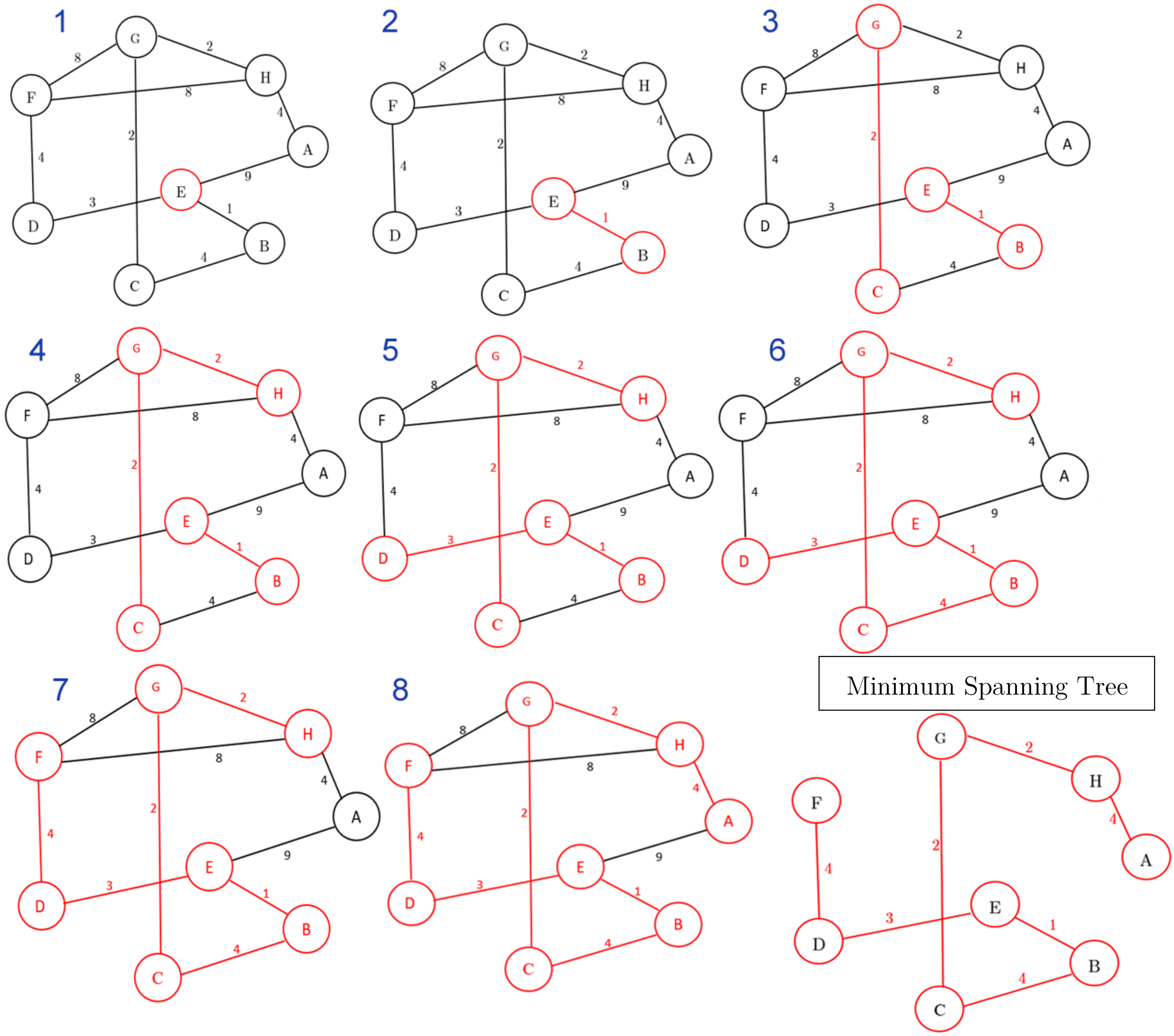


Cost $= C_{E,B} + C_{E,D} + C_{B,C} + C_{C,G} + C_{G,H} + C_{D,F} + C_{H,A}$

$= 1 + 3 + 4 + 2 + 2 + 4 + 4 = 20$

Thrown Edges due to cycle $= e_{F,G},\ e_{F,H},\ e_{A,E}$

Question 4. Kruskal's Algorithm



Minimum Spanning Tree

We are taking the edges with the minimum costs, if the vertices in the edge we took belongs to distinct trees we merge them, if not we do not take this edge. This process continues until there is only one tree left.

Thrown Edges due to cycle = $e_{F,G}$, $e_{F,H}$, $e_{A,E}$

Trace & Cost

1. v_E
2. $e_{E,B}$ (1)
3. $e_{C,G}$ (2)
4. $e_{G,H}$ (2)
5. $e_{E,D}$ (3)
6. $e_{C,B}$ (4)
7. $e_{D,F}$ (4)
8. $e_{H,A}$ (4)

Question 5. Breadth-First Search

Trace Steps:

1. Start from vertex G.
   Visited Vertices S. Enqueue G. Set $(d_G,p_G) = 0$
2. Visit unvisited children of G which are (F,C,H). Enqueue {F,C,H}.
   Set $(d_F,p_F) = (d_C,p_C) = (d_H,p_H) = (1,G)$ and S = {G}
3. Dequeue F. Enqueue its unvisited children D.
   Set $(d_D,p_D) = (2,F)$ and S = {G,F}
4. Dequeue H. Enqueue its unvisited children A.
   Set $(d_A,p_A) = (2,H)$ and S = {G,F,H}
5. Dequeue C. Enqueue its unvisited children B.
   Set $(d_B,p_B) = (2,C)$ and S = {G,F,H,C}
6. Dequeue D. Enqueue its unvisited children E.
   Set $(d_E,p_E) = (3,D)$ and S = {G,F,H,C,D}
7. Dequeue A. No unvisited children.
   Set S = {G,F,H,C,D,A}
8. Dequeue B. No unvisited children.
   Set S = {G,F,H,C,D,A,B}
9. Dequeue E. No unvisited children.
   Set S = {G,F,H,C,D,A,B,E}

Shortest Path Table

A	B	C	D	E	F	G	H
(2,H)	(2,C)	(1,G)	(2,F)	(3,D)	(1,G)	(0,G)	(1,G)

* Visited Vertices = S
* (d_X,p_Y) = (distance to vertex X, previous vertex was Y)

Question 6. Topological ordering

Trace Steps:

1. S = {G}. Take G, print then remove.
2. S = {D,H}. Take D randomly. Print then remove.
3. S = {H,A}. Take H randomly. Print then remove
4. S = {A}. Take A, print then remove.
5. S = {B,E}. Take B randomly. Print then remove
6. S = {E}. Take E, print then remove.
7. S = {I}. Take I, print then remove.
8. S = {F}. Take F, print then remove.
9. S = {C}. Take C, print then remove.
10. S = {t}. Take t, print then remove.

Output: s, G, D, H, A, B, E, I, F, C, t.

* S = In-Degree 0 vertices set