VE203 Discrete Math
Spring 2022 — HW6 Solutions

April 19, 2022

Exercise 6.1

Let us first determine the set of vertices and set of edges of the first graph:

$V_1 = \{a, b, c, d, e, f, g, h, i, j\}$
$E_1 = \{(a, b), (b, c), (c, d), (d, e), (e, a), (b, g), (c, h), (d, i), (e, j), (a, f), (g, i), (g, j), (h, f), (h, j), (f, i)\}$

Let us determine the set of vertices and set of edges of the second graph:

$V_2 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$
$E_2 = \{(1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 1), (3, 7), (6, 7), (2, 8), (5, 8), (1, 9), (4, 9), (0, 7), (0, 8), (0, 9)$

Let us determine the set of vertices and set of edges of the third graph:

$V_3 = \{\alpha, \beta, \gamma, \delta, \varepsilon, \zeta, \lambda, \eta, \iota, \kappa\}$
$E_3 = \{(\alpha, \beta), (\beta, \gamma), (\gamma, \delta), (\delta, \varepsilon), (\varepsilon, \zeta), (\zeta, \lambda), (\lambda, \kappa), (\eta, \iota), (\iota, \kappa), (\alpha, \varepsilon), (\zeta, \eta), (\kappa, \alpha), (\delta, \iota), (\lambda, \gamma), (\beta,$

By comparing the two sets of edges, we can define the following one-to-one and onto function f from V_1 to V_2.

$$
\begin{aligned}
f(a) &= 1 \\
f(b) &= 2 \\
f(c) &= 8 \\
f(d) &= 5 \\
f(e) &= 6 \\
f(f) &= 9 \\
f(g) &= 3 \\
f(h) &= 0 \\
f(i) &= 4 \\
f(j) &= 7
\end{aligned}
$$

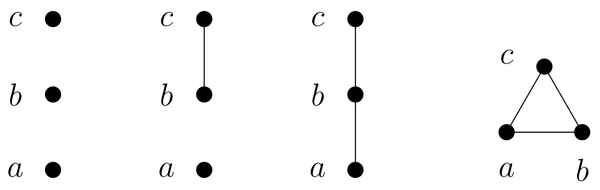
By comparing the two sets of edges, we can define the following one-to-one and onto function f from V_2 to V_3.

$$
\begin{aligned}
g(1) &= \alpha \\
g(2) &= \beta \\
g(3) &= \gamma \\
g(4) &= \delta \\
g(5) &= \iota \\
g(6) &= \kappa \\
g(7) &= \lambda \\
g(8) &= \eta \\
g(9) &= \varepsilon \\
g(10) &= \zeta
\end{aligned}
$$

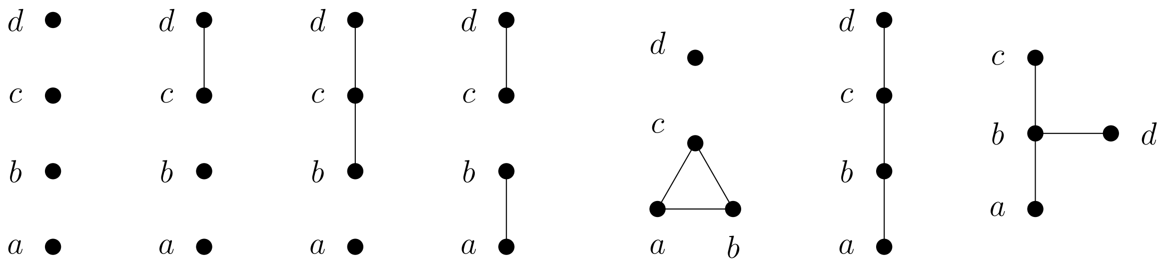
From first graph to the third graph:
$h(x) = g(f(x))$

Exercise 6.2

i)



ii)



1 2 3 4 5 6 7

8 9 10 11

In this problem, all pairs of complement graphs are:

1 and 11
2 and 10
3 and 9
4 and 8
5 and 7
6 and 6

in which 6 is self-complementary.

Exercise 6.3

Notice that this is a symmetric statement, hence we only need to prove one direction, then the other direction follows directly. More specifically, since a complement of a graph is still a graph, we just insert $\bar{G}$ and $\bar{H}$ as our original graphs G, H for one direction then we are done.

Now we prove that if two simple graphs are isomorphic, then their complement graphs are isomorphic as well. Without losing of generality, we may assume that G, H are finite.

Since G, H are isomorphic, then we know there exist a function f such that

$$f : V(G) \to V(H)$$

with

$$uv \in E(G) \Leftrightarrow f(u)f(v) \in E(H)$$

for $u, v \in V(G)$. Now we consider the complement of G and H, namely $\bar{G}$ and $\bar{H}$. We now show that f mentioned above is exactly the isomorphism between $\bar{G}$ and $\bar{I}$. Firstly, we know that

$$V(G) = V(\bar{G}), \quad V(H) = V(\bar{H})$$

and

$$\underset{u,v \in V(G)}{\forall} uv \notin E(G) \Leftrightarrow uv \in E(\bar{G}), \qquad \underset{u,v \in V(H)}{\forall} uv \notin E(H) \Leftrightarrow uv \in E(\bar{H}).$$

Furthermore, we also see that the contrapositive argument for f, which can be expressed as

$$\underset{u,v \in V(G)}{\forall} uv \notin E(G) \Leftrightarrow f(u)f(v) \notin E(H),$$

can lead to the following result:

$$uv \in E(\bar{G}) \Leftrightarrow uv \notin E(G) \Leftrightarrow f(u)f(v) \notin E(H) \Leftrightarrow f(u)f(v) \in E(\bar{H})$$

for $u, v \in V(G)$. Equivalently,

$$uv \in E(\bar{G}) \Leftrightarrow f(u)f(v) \in E(\bar{H})$$

for $u, v \in V(G)$ as we expected.

Exercise 6.4

We only need to consider the case that $n \geq 3$, since K_n is simple, for $n < 2$, there can't be any cycles in $K_n, n < 2$ clearly.

Now, for any $K_n, n \geq 3$, since it is complete, we can get a cycle by choosing any order of the sequence

$$(1, 2, 3, \ldots, n)$$

which represents an orientation and the traversal path of the cycle. Noting that this is unique up to a circulation permutation sense, since a cycle will have a same orientation with different ending point and starting point, as long as its circulation permutation sense is the same. Therefore, we can get the number of circulation permutation as

$$\frac{n!}{n}$$

by first total permutating them and divide by the repeated result for circulation cases, which has n of them.

Furthermore, since we are considering an undirected graph, hence, the orientation is gone as well. This further leads to the fact that the number of cycles of length n in the complete graph $K_n, n \geq 3$ be given by

$$\frac{n!}{n} \times \frac{1}{2} = \frac{(n-1)!}{2}.$$

In all, we have

$$\begin{cases} 0 & , 0 \leq n \leq 2 \\ \frac{(n-1)!}{2} & , n \geq 3 \end{cases}$$

Exercise 6.5

To prove this, we build a graph with a total of n vertices. Now we have at least one component for the graph, and the total number of edges can be found as $\sum_i \left(\begin{array}{c} n_i \\ 2 \end{array} \right)$, where n_i is the number of vertices in the i th component.

On the other hand, we have the number of edges in K_n as $\left(\begin{array}{c} n \\ 2 \end{array} \right)$. Now since a complete graph has the maximum number of edges among graphs with the same number of vertices, we have $\left(\begin{array}{c} n \\ 2 \end{array} \right) \geq \sum_i \left(\begin{array}{c} n_i \\ 2 \end{array} \right)$.

Exercise 6.6

If G does not contain any cycle, then we look at any one vertex in this graph. Since it has at least two edges connected and it cannot connect with itself without forming a cycle, it will connect to two other vertices in the graph. And then we can expand this to the two new vertices, and so on, until all the vertices are reached in this walk. If any vertex appears more than once in this walk, then we have a cycle, which is a contradiction. If all vertices are distinct, we look at the first and last vertices in this walk. In order not to form a cycle they cannot be connected to any other vertices, but this also means they have a degree of 1 , leading to a contradiction. Therefore, G contains a cycle.

Exercise 6.7

i) For a k-regular bipartite graph B_k with $k \geq 2$, we know that $V\left(B_k\right)$ can be partitioned into X and Y such that

$$X \cap Y = \varnothing, \quad X \cup Y = V\left(B_k\right)$$

with the fact that for all edge $ab \in E\left(B_k\right)$ with starting point a and ending point b, a and b can't both be in the same set of X or Y.

Now, since we know that $k \geq 2$, which means that for all $x \in X$, there is k different vertices in Y connected with x, resulting the fact that there can't exist a cut-edge in B automatically. We can see this by the fact that for all $e \in E\left(B_k\right), B_k - e$ is still connected since there are always at least $k-1$ edges starting in X and ending in Y for every $x \in X$ (or $y \in Y$, equivalently), together with

$$k \geq 2 \Leftrightarrow k-1 \geq 1,$$

we know there is at least 1 such edge exists.

ii) We first note that if for a $k-$ regular bipartite graph B_k, there exists a matching, it needs to be perfect. We see it from several steps. Firstly, since for a bipartite graph, we can partition its vertex set $V\left(B_k\right)$ into X and Y as in (i), then we first show that $|X| = |Y|$. Since for every $x \in X$, there are k edges starting from it and ending in Y, we know that the total number of edges in B_k is $k \cdot |X|$. Same for Y, hence we see that the number of edges in B_k is also equal to $k \cdot |Y|$. Since $k \geq 1$, we see that

$$k \cdot |X| = k \cdot |Y| \Rightarrow |X| = |Y|.$$

Now, if there exists a matching in B_k, then every edge in the matching is starting from X and ending in Y since it is a bipartite graph. Furthermore, since between two edges in the matching, they can't have a same vertex among their end points, together with the fact that $|X| = |Y|$, if this matching exists, then every vertex will be paired in an edge. Now, from Hall's Theorem, we first prove the following:

$$\underset{A \subseteq X}{\forall} |N(A)| \geq |A|.$$

We can show it as follows. Firstly, we note that for every $A \subseteq X, N(A) \geq k$ because of the property of k-regular, namely for every $a \in A$, its k edges are all starting from a ending with same k vertices in Y. Then, we only have to prove that there can't exist a subset $A \subseteq X$ such that

$$k = |N(A)| < |A|,$$

since if this is true, then for every $|N(A)| \geq k$, this A can not exist. We see it as follows. If there are such A with $|N(A)| = k < |A|$, then we know that $\underset{a \in A}{\forall} \underset{b \in N(A)}{\forall} \underset{e \in B_k}{\exists} e$ starts from a ends at b. But this implies that

$$\underset{b \in N(A)}{\forall} \deg(y) = |A| > k,$$

which is a contradiction. Therefore, we see that there exists such a matching in B_k for $k \geq 1$. With the fact that every matching in B_k is a perfect matching, we are done.

iii)

Exercise 6.8

We see this is true from G is a tree $\Leftrightarrow$ G is connected and contains no cycles. Noting that if G contains no cycles, then every edge e in G is a cut-edge since there are no cycles contain e, which is true because there are no cycles in the first place. Therefore, we further have G is a tree $\Leftrightarrow$ G is connected and e is a cut-edge for all $e \in E(G)$.

Exercise 6.9

Since $a \in A, A - a \subset A$, so $A - a \in I$. Since $b \in B - A, b \in B - (A - a)$ and $|A - a| < |B|$, so $(A - a) \cup b \in I$. Then, since $|(A - a) \cup b| = |A| = |B|, (A - a) \cup b$ is maximal and is a basis.

Exercise 6.10

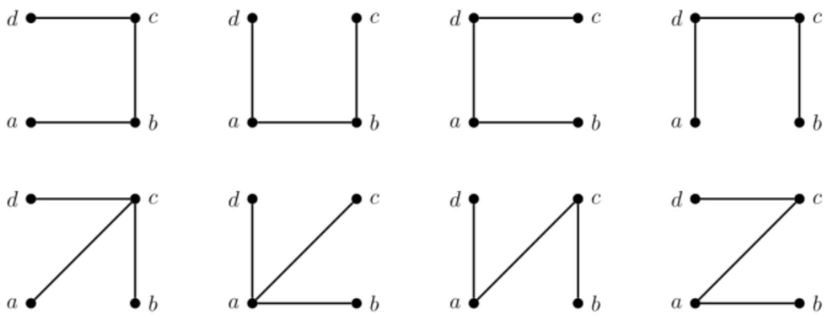
(I1): since $|\phi| = 0 \leq k$ and $\phi \subset E, \phi \in I$.

(I2): if $A \in I$ and $B \subset A, |B| \leq |A| \leq k \Rightarrow B \in I$.

(I3): if $A, B \in I$ and $|A| < |B|$. Suppose $b \in B - A$. Then $|A \cup b| \leq |B| \leq k$, so $A \cup b \in I$.

Exercise 6.11

Given the following graph we can draw the following 8 spanning trees drawn as



Exercise 6.12

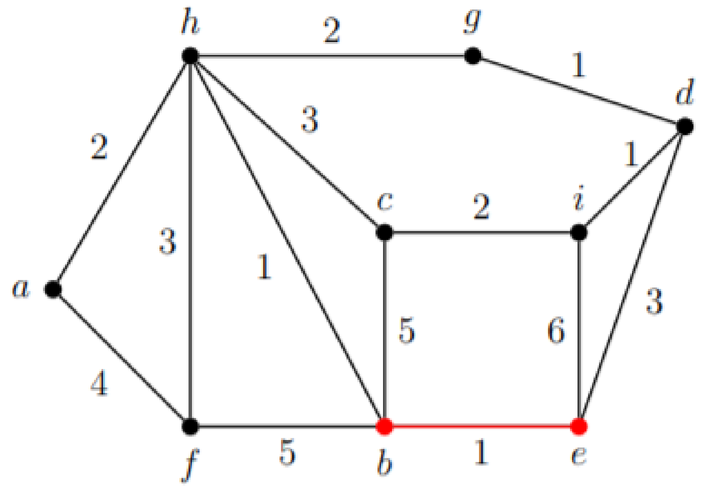
i)

For Kruskal's algorithm, we first list the edges in the increasing order of its weighted length, which is given by

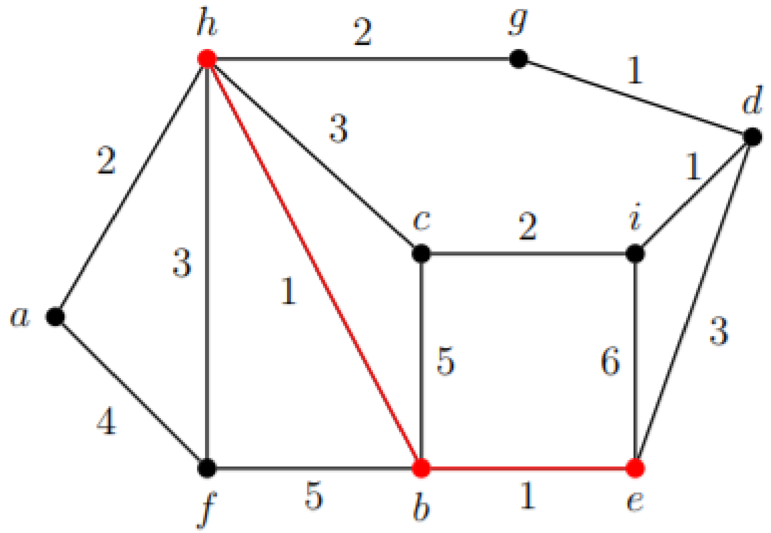
$$1, 1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 5, 5, 6$$

The ties don't matter for the result(or one can say for different order of tied edges will give a specific minimum-weight spanning tree). We randomly choose the following order of edge and list as $be, bh, id, dg, ah, hg, ci, hf, hc, ed, af, fb, bc, ie$.

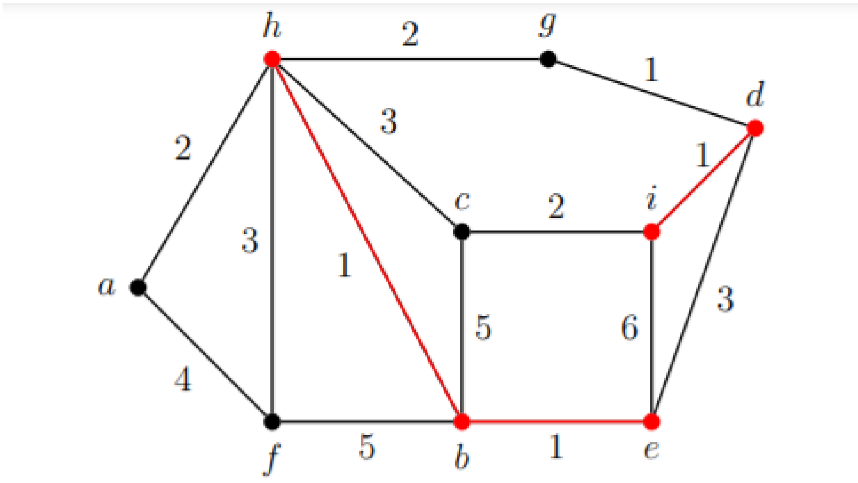
Now, we draw the following spanning tree by Kruskal's algorithm. First, we add *be* in our graph



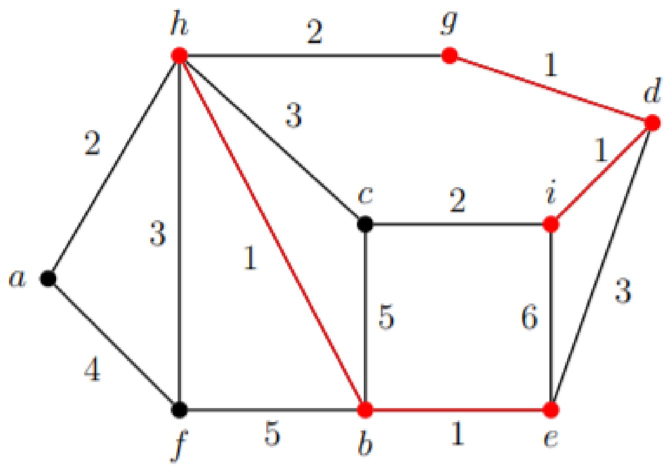
Next, we see that adding *bh* will not introduce a cycle, hence we add it in our graph



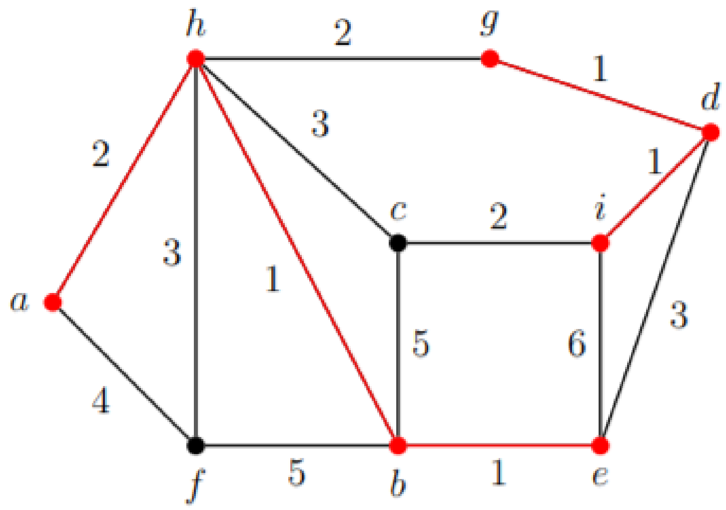
Next, we see that adding id will not introduce a cycle, hence we add it in our graph



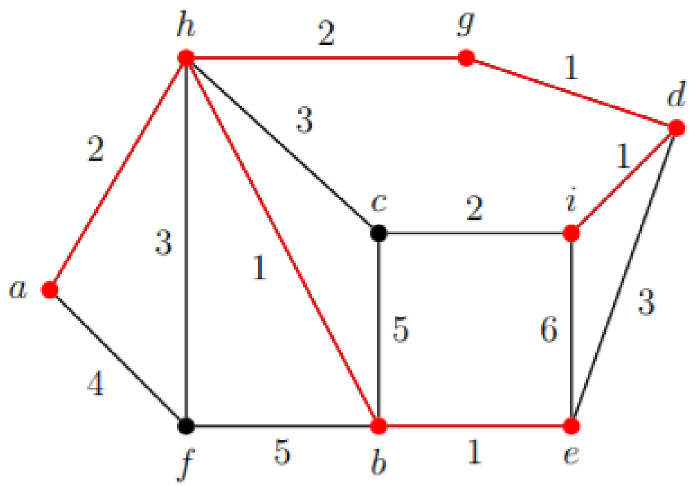
Next, we see that adding dg will not introduce a cycle, hence we add it in our graph



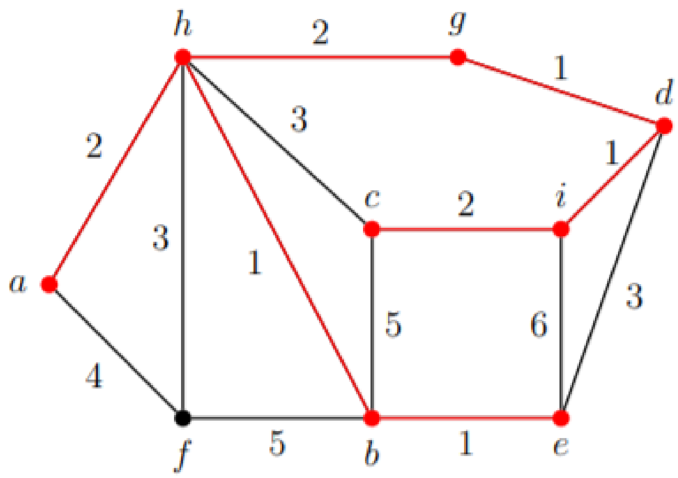
Next, we see that adding ah will not introduce a cycle, hence we add it in our graph



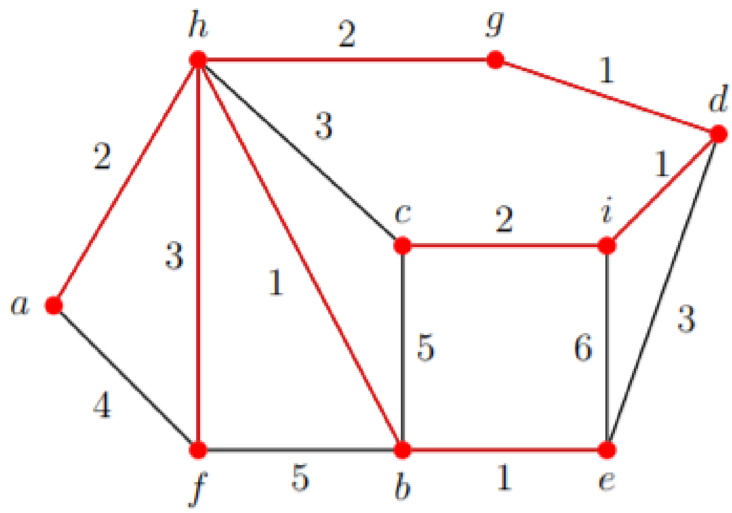
Next, we see that adding hg will not introduce a cycle, hence we add it in our graph



Next, we see that adding ci will not introduce a cycle, hence we add it in our graph



Next, we see that adding hf will not introduce a cycle, hence we add it in our graph

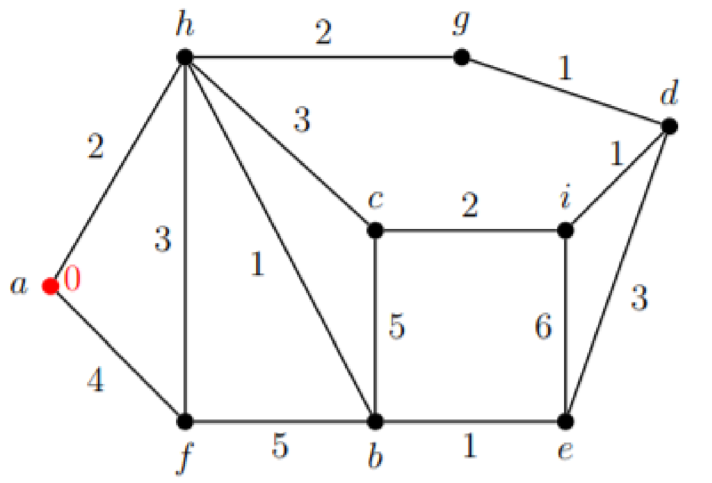


Next, we see that adding hc will introduce a cycle, hence we omit it. Also, adding ed, af, fb, bc or ie will all introduce a cycle, hence we terminate the algorithm, resulting a desired minimum-weight spanning tree.

ii)

Given the root vertex a, find a shortest-path spanning tree via Dijkstra's algorithm. List the edges chosen in order, list the shortest path distance (from root vertex) to each vertex. Sketch the tree.

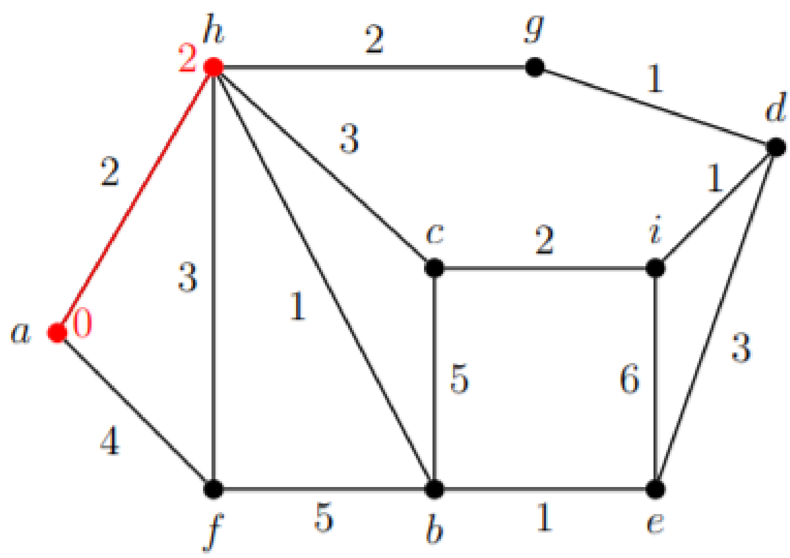
Solution:



We first find the neighbors of a and the temporary cost of them. The neighbors are h and f, with the temporary cost

$$\text{Cost}_{\ a\to h} = 2,$$
$$\text{Cost}_{\ a\to f} = 4.$$

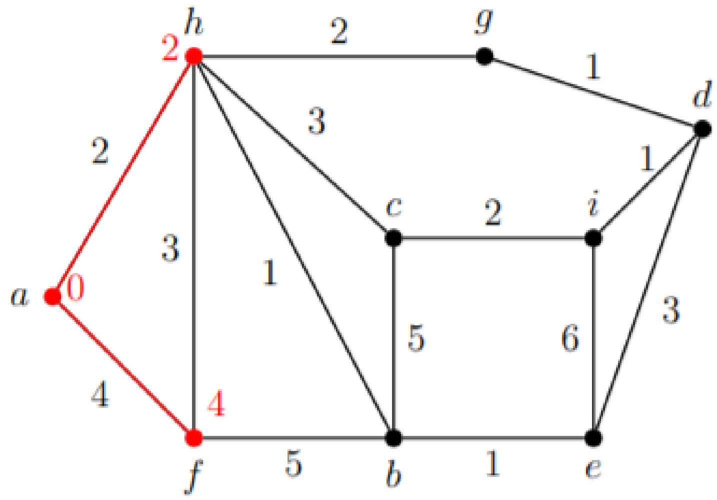
We choose the minimal costed edge, which is ah and add it in the graph where we specify the determined cost from the root a to the node beside it in red.



We first find the neighbors of a, h and the temporary cost of them. The neighbors are f, g and c, with the temporary cost

$$\begin{aligned}
&\text{Cost}_{\ a\to f} = 4,\\
&\text{Cost}_{\ h\to c} = 2+3 = 5,\\
&\text{Cost}_{\ h\to g} = 2+2 = 4.
\end{aligned}$$

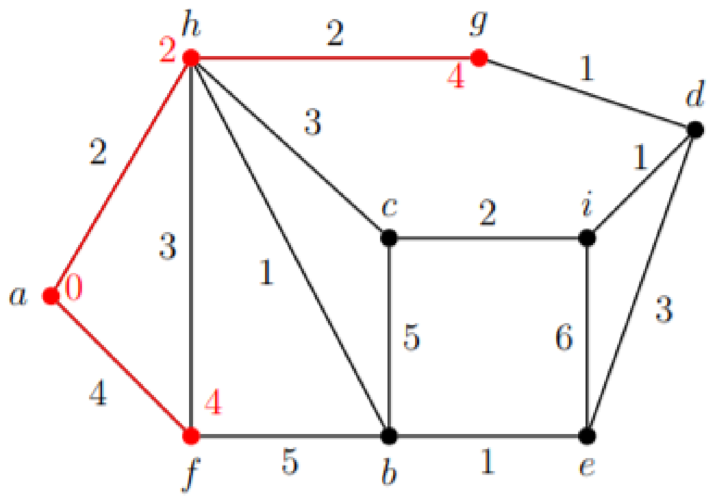
There is a tie, and we randomly choose one of which. In this case, we choose af to be added in the graph



Next, we consider the neighbors of h, f and the temporary cost of them. The neighbors are c, b and g, with the temporary cost

$$\begin{aligned}
\text{Cost}_{\ h\to c} &= 2+3=5,\\
\text{Cost}_{\ h\to g} &= 2+2=4,\\
\text{Cost}_{\ f\to b} &= 4+5=9.
\end{aligned}$$

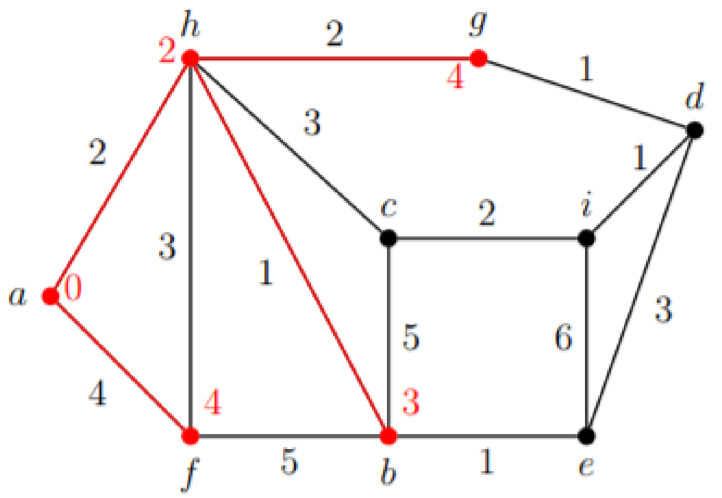
We choose the minimal costed edge, which is hg and add it in the graph



Next, we consider the neighbors of h, f and g and the temporary cost of them. The neighbors are b, c and d, with the temporary cost

$$\begin{aligned}
\text{Cost }_{h\to b} &= 2+1=3,\\
\text{Cost }_{f\to b} &= 4+5=9,\\
\text{Cost }_{h\to c} &= 2+3=5,\\
\text{Cost }_{g\to d} &= 4+1=5.
\end{aligned}$$

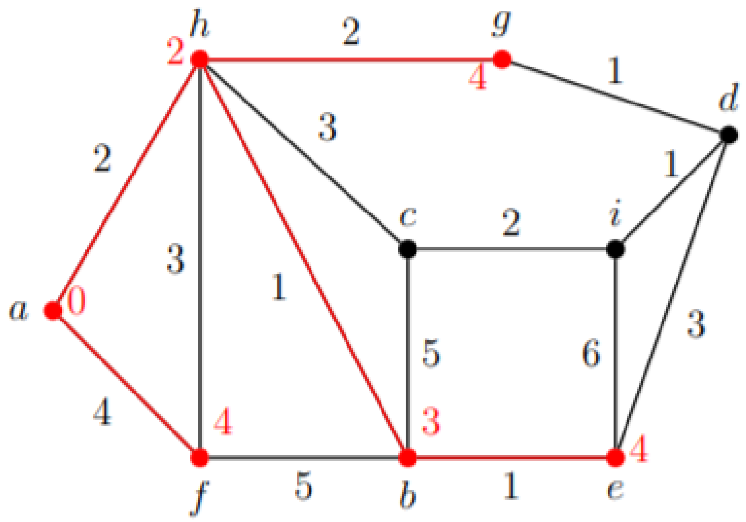
We choose the minimal costed edge, which is hb and add it in the graph



Next, we consider the neighbors of h, b and g and the temporary cost of them. The neighbors are c, e and d, with the temporary cost

$$\begin{aligned}
\text{Cost }_{h\to c} &= 2+3=5 \\
\text{Cost }_{b\to c} &= 3+5=8 \\
\text{Cost }_{b\to e} &= 3+1=4 \\
\text{Cost }_{g\to d} &= 4+1=5.
\end{aligned}$$

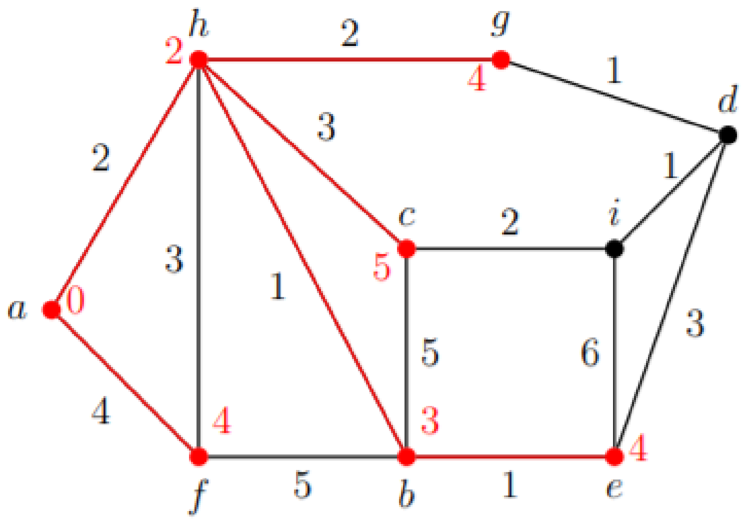
Notice that we do not need to consider f anymore, since any edges expanded from which will introduce a cycle. We choose the minimal costed edge, which is be and add it in the graph



Next, we consider the neighbors of h, b, e and g and the temporary cost of them. The neighbors are c, i and d, with the temporary cost

$$\begin{aligned}
\text{Cost}_{\ h\to c} &= 2+3=5\\
\text{Cost}_{\ b\to c} &= 3+5=8\\
\text{Cost}_{\ e\to i} &= 4+6=10\\
\text{Cost}_{\ g\to d} &= 4+1=5\\
\text{Cost}_{\ e\to d} &= 4+3=7
\end{aligned}$$

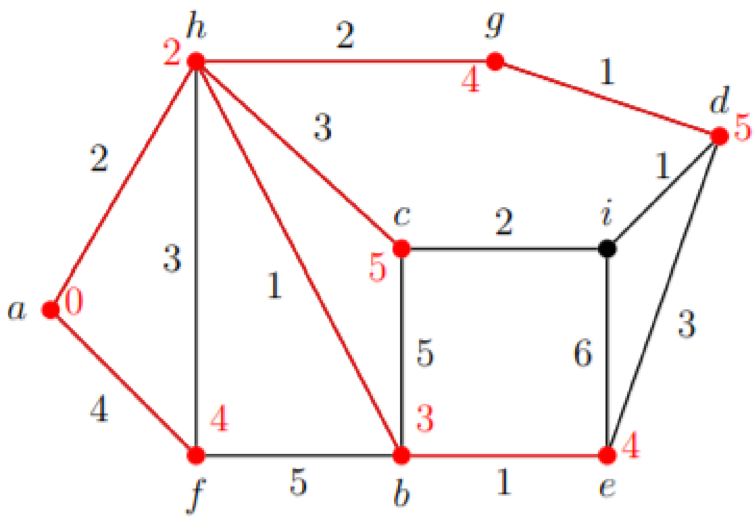
There is a tie, and we randomly choose one of which. In this case, we choose hc to be added in the graph



Next, we consider the neighbors of c, e and g and the temporary cost of them. The neighbors are i and d, with the temporary cost

$$\begin{aligned}
\text{Cost}_{\ c\to i} &= 5 + 2 = 7, \\
\text{Cost}_{\ e\to i} &= 4 + 6 = 10, \\
\text{Cost}_{\ g\to d} &= 4 + 1 = 5, \\
\text{Cost}_{\ e\to d} &= 4 + 3 = 7.
\end{aligned}$$

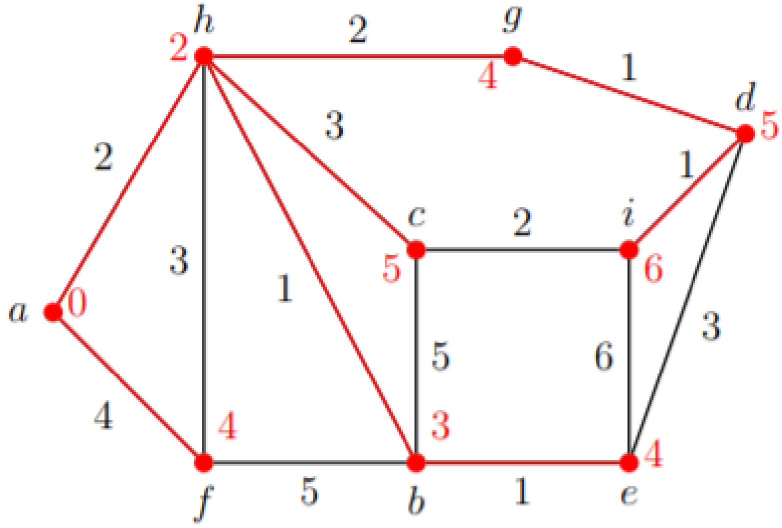
Notice that we do not need to consider b anymore, since any edges expanded from which will introduce a cycle. We choose the minimal costed edge, which is gd and add it in the graph



Finally, we consider the neighbors of c, e and d and the temporary cost of them. The only neighbor left is i, with the temporary cost

$$\begin{aligned} \text{Cost}_{\ c\to i} &= 5+2=7 \\ \text{Cost}_{\ d\to i} &= 5+1=6, \\ \text{Cost}_{\ e\to i} &= 4+3=7 \end{aligned}$$

We choose the minimal costed edge, which is di and add it in the graph



Follow the procedure, one can see the order to choose the vertex adding in the graph, and also the red value labeled beside each vertex indicates the shortest path distance from root vertex a to it.