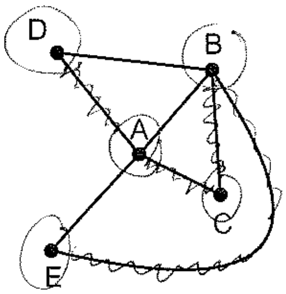
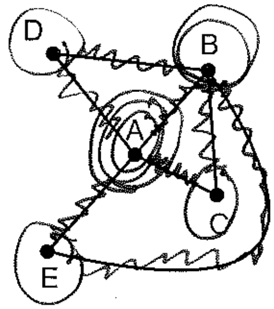
CHAPTER 2 – BUSINESS EFFICENCY

A path that visits every vertex exactly once is a ***Hamiltonian path***.
A circuit that visits every vertex exactly once is a ***Hamiltonian circuit***.



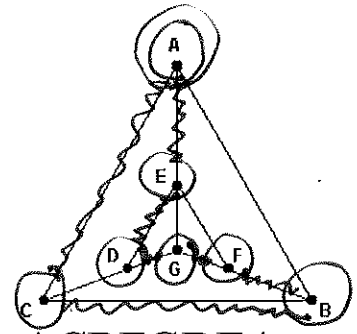
EBCAD

Hamiltonian path



ADBACBEA

Euler circuit

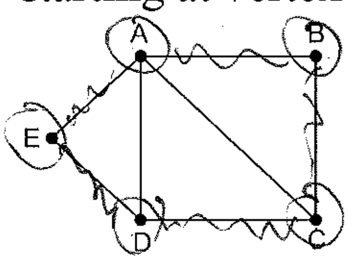


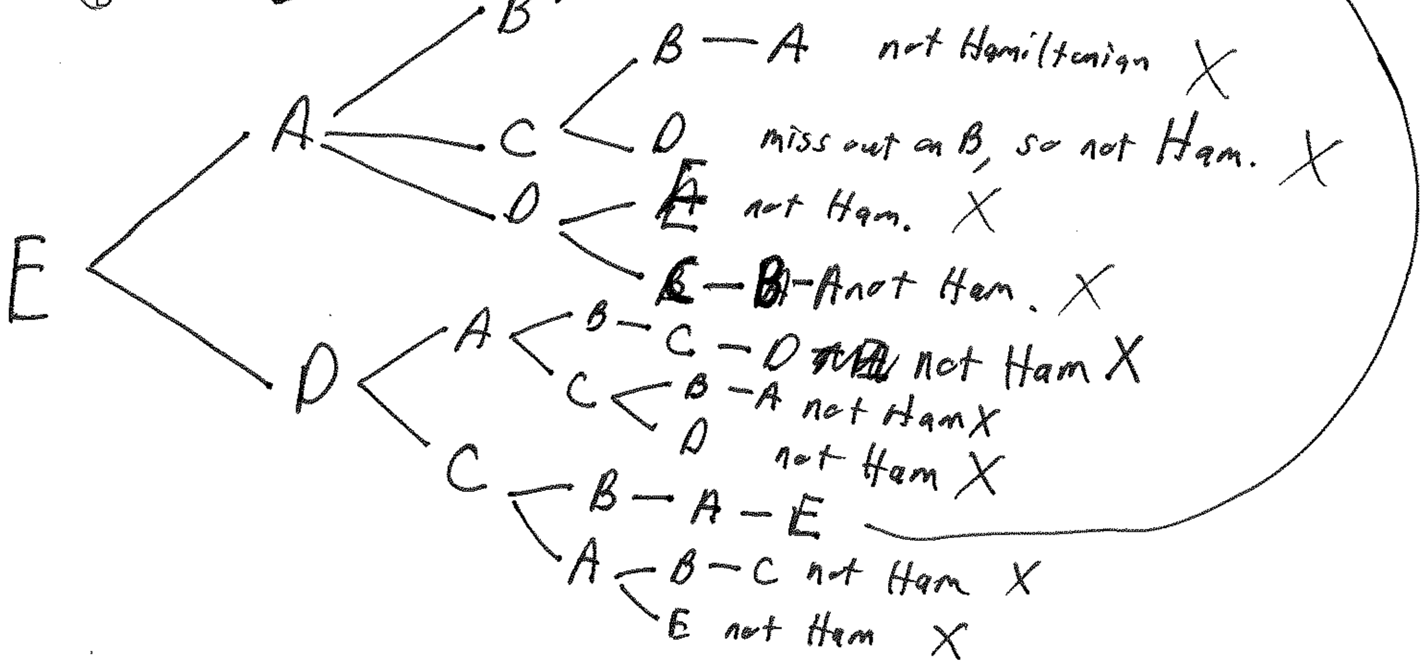
ACBFGDEA

Hamiltonian circuit

Use the method of trees (brute Force) to find all Hamiltonian circuits in the graph below starting at vertex E.

One unique Hamiltonian circuit





The ***Traveling Salesman Problem*** (***TSP***) is where a least cost Hamiltonian circuit is found.

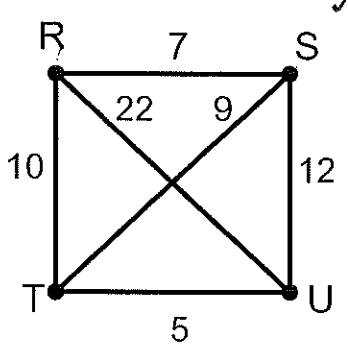
The graph below shows the time in minutes to travel between the vertices R, S, T, and U. Find the least cost Hamiltonian circuit for this graph starting at vertex T.

$3\cdot 2\cdot 1$ = routes that visit each V once

$\frac{3\cdot 2\cdot 1}{2}$ = unique Ham circuits

Least cost is 34 min

TRSUT or TUSRT



T —10— R —7— S —12— U —5— T 34 min

T —10— R —22— U —12— S —9— T 53 min

T —9— S —7— R —22— U —5— T 43 min

T —9— S —12— U —22— R —10— T 53 min

T —5— U —22— R —7— S —9— T 43 min

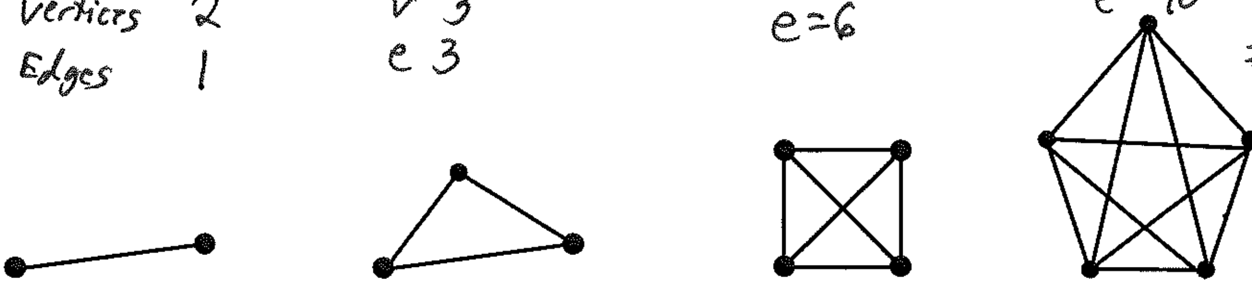
T —5— U —12— S —7— R —10— T 34 min

A ***complete*** graph is a graph in which every pair of vertices is connected by exactly one edge. How are the number of vertices and the number of edges related in a complete graph?

Vertices 2 v 3 v=4 V=5

Edges 1 e 3 e=6 e=10

#edges $= \frac{V(V-1)}{2}$



$\frac{5(4)}{2} = 10$

Brute Force Method:

$n = \#$ vertices

- There are $n!$ routes that visit every vertex once in a complete graph.
- There are $n!/2$ unique Hamiltonian circuits in a complete graph.
- If a starting point is specified, there are $(n-1)!/2$ unique Hamiltonian circuits in a complete graph.

Remember that $n! = n \times (n-1) \times (n-2) \times \ldots \times 3 \times 2 \times 1$

Consider a complete graph with 14 vertices.

a) How many edges does the graph have?

$\#\text{ edges} = \frac{V(V-1)}{2} = \frac{14(13)}{2} = 91$ edges

b) If you can start at any vertex, how many different routes are there if every vertex is visited exactly once?

$n! \quad 14! = 14 \cdot 13 \cdot 12 \cdot \ldots \cdot 1 = 87{,}178{,}291{,}200$

c) If you can start at any vertex, how many different Hamiltonian circuits are there?

$\frac{n!}{2} \quad \frac{14!}{2} = 43{,}589{,}145{,}600$

d) If you start at a given vertex how many different Hamiltonian circuits are there?

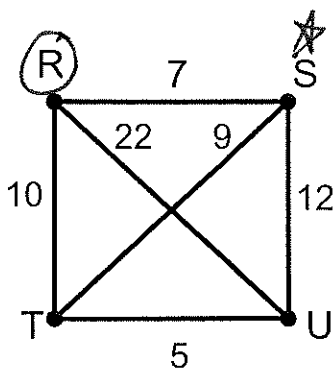
$\frac{(n-1)!}{2} \quad \frac{(14-1)!}{2} = \frac{13!}{2} = \frac{13 \cdot 12 \cdot 11 \cdot \ldots \cdot 1}{2} = 3{,}113{,}510{,}400$

A ***heuristic algorithms*** is an algorithm that is fast but may not be optimal. A ***greedy algorithm*** is one in which the choices are made by what is best at the next step.

Nearest Neighbor Algorithm:

Starting from the home city, visit the nearest city first. Then visit the nearest city that has not already been visited. Return to the home city when no other choices remain.

The graph below shows the time in minutes to travel between the vertices R, S, T, and U. Find the least cost Hamiltonian circuit for this graph using the nearest neighbor algorithm starting at R. Is the result different if you start at S? Yes

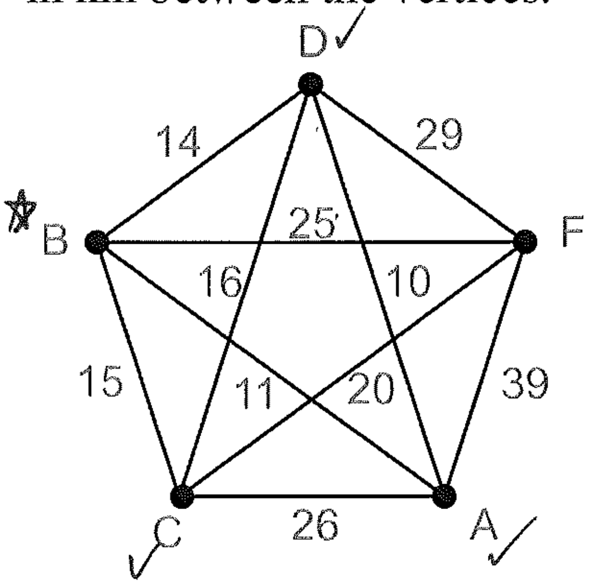


$R \xrightarrow{7} S \xrightarrow{9} T \xrightarrow{5} U \xrightarrow{22} R = 43 \text{ min}$

$S \xrightarrow{7} R \xrightarrow{10} T \xrightarrow{5} U \xrightarrow{12} S = 34 \text{ min}$

Use the nearest neighbor algorithm to find the shortest Hamiltonian circuit for the graph below starting at B. The values on the edges are the distance in km between the vertices.

$B \xrightarrow{11} A \xrightarrow{10} D \xrightarrow{16} C \xrightarrow{20} F \xrightarrow{25} B = 82$ km



The chart below shows the distance between vertices in miles. Find the shortest Hamiltonian circuit using the nearest neighbor algorithm starting at vertex C

	A	B	C	D	E	F
A	0	26	49	50	7	34
B	26	0	48	24	28	36
C	49	48	0	18	40	15
D	50	24	18	0	13	17
E	7	28	40	13	0	20
F	34	36	15	17	20	0

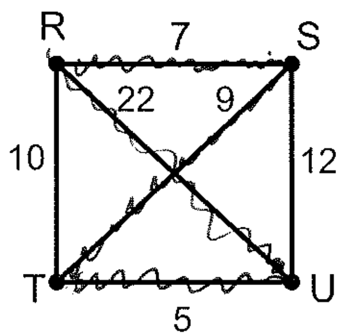
$C \xrightarrow{15} F \xrightarrow{17} D \xrightarrow{13} E \xrightarrow{7} A \xrightarrow{26} B \xrightarrow{48} C$

126 miles

Sorted Edges Algorithm:

1. Arrange edges of the complete graph in order of increasing cost
2. Select the lowest cost edge that has not already been selected that
   a. Does not cause a vertex to have 3 edges
   b. Does not close the circuit unless all vertices have been included.

Find the shortest Hamiltonian circuit using the sorted edges algorithm for the graphs below.

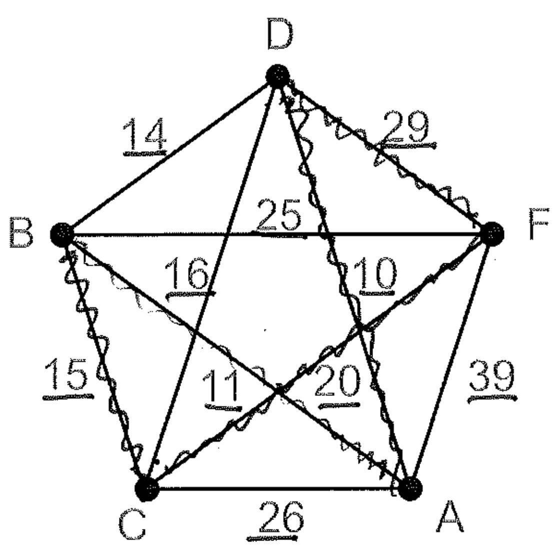


5 ✓ 7 ✓ 9 ✓ 10 X 12 X 22 ✓

10: Causes T to have 3 edges and close ~~loop~~ circuit

12: Causes S to have 3 edges and close ~~loop~~ circuit

TURST = 43min

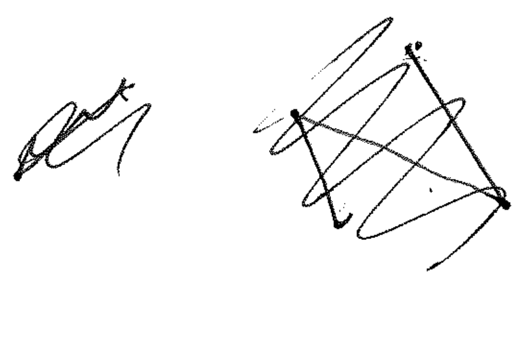


10 ✓ 11 ✓ 14 X 15 ✓ 16 X 20 ✓ 25 X 26 X 29 39

14: Close ~~loop~~ circuit

16: Close ~~loop~~ circuit

DABCFD = 85 km

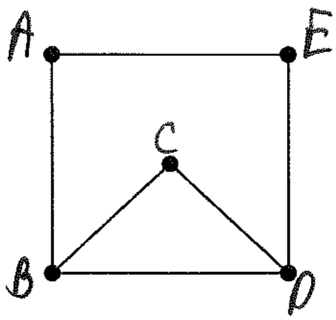


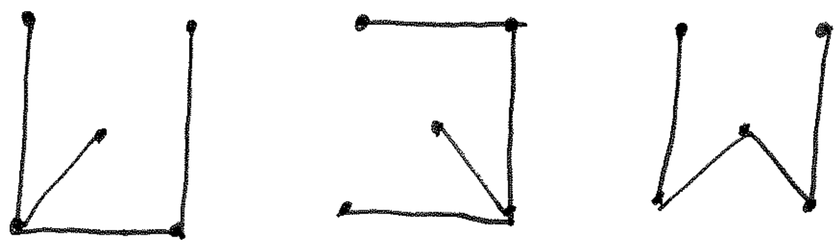
A connected graph that has no circuits is a ***tree***. A ***spanning tree*** is a tree that has all the vertices of the original graph.

To create a spanning tree from a graph,

1. Find a circuit and remove one edge
2. Continue until there are no circuits

Create 3 different spanning trees from the graph below:

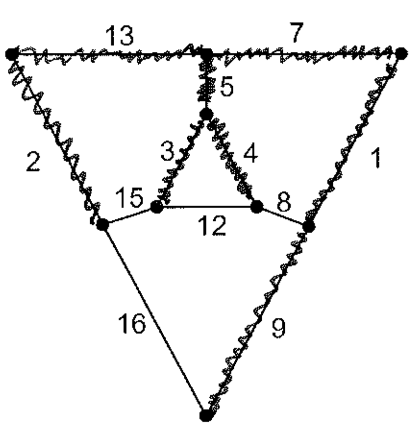


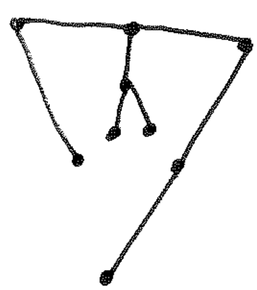


A ***minimal spanning tree*** is a spanning tree with the smallest possible weight.

Kruskal's Algorithm:
Add edges in order of increasing cost so that no circuit is formed.

Use Kruskal's algorithm to find the minimal spanning tree for the graph below. What is the cost?





Cost is 44 units

A list of vertices connected by arrows is a ***directed graph*** or ***digraph***.

If the tasks cannot be completed in a random order, then the order can be specified in an ***order-requirement digraph***.

If the time to complete a task is shown on the digraph, it is a ***weighted*** digraph.

An ***independent task*** is one that can be done independently of any of the other tasks.

Pizza Dinner

To make a pizza, you need to pre-heat the oven, unwrap the frozen pizza, and bake it for 15 minutes. You also need to find plates. Show these tasks in a weighted order requirement digraph.

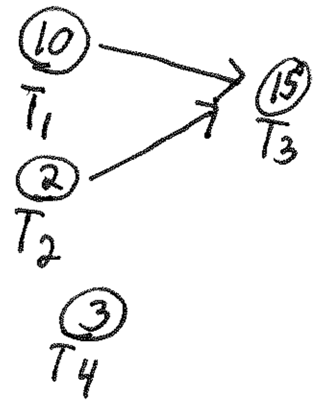
T_1 = Pre-heat oven = 10 min

T_2 = unwrap pizza = 2 min

T_3 = Bake = 15 min

T_4 = Find plates = 3 min

Made up times



Are any of the tasks independent? T_4

How long to make the pizza if one person is available (no multi-tasking)?

$10 + 2 + 15 + 3 = 30$ min

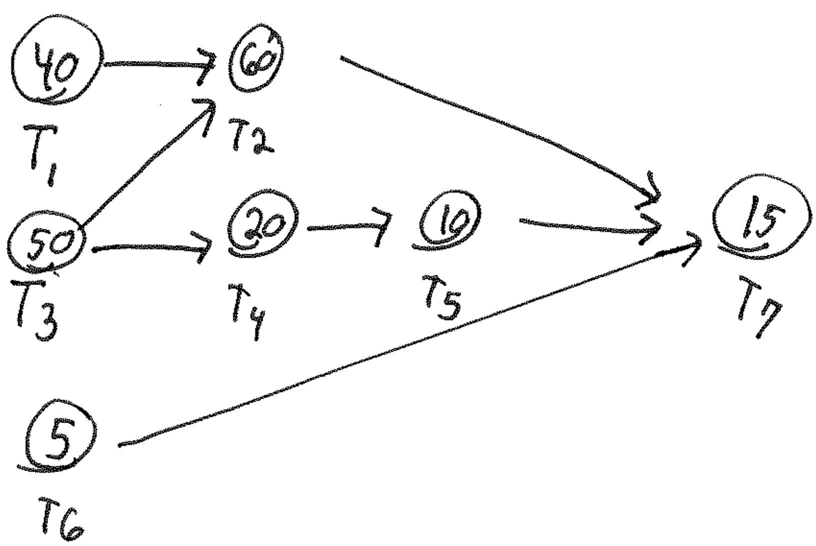
How long to make the pizza if two people are available?

25 min   Critical path T_1 T_3 = 25 min

Yard Work

Arrange the tasks below in a weighted order requirement digraph.

- T_1: Trim edges (40 minutes)
- T_2: Mow (60 minutes)
- T_3: Weed flower beds (50 minutes)
- T_4: Plant flowers (20 minutes)
- T_5: Mulch flowers (10 minutes)
- T_6: Refill bird feeder (5 minutes)
- T_7: Sweep (15 minutes)



A ***critical path*** on the diagraph is the path that determines the earliest completion time.

What is the critical path for making the pizza?

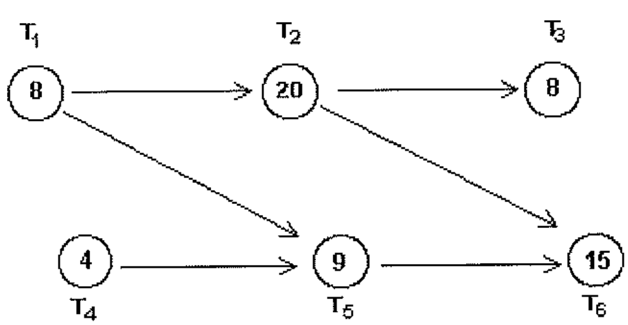
$T_1 \to T_3$ 25 min

What is the critical path for yard work?

$T_3\ T_2\ T_7 = 125$ min

What is the critical path in the digraph below?

time in min



$T_1\ T_2\ T_3 = 36$ min

$T_1\ T_2\ T_6 = 43$ min critical path

$T_4\ T_5\ T_6 = 28$ min

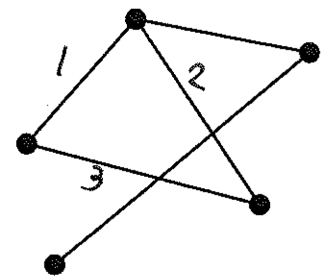
$T_1\ T_5\ T_6 = 32$ min

SAMPLE EXAM QUESTIONS FROM CHAPTER 2

How many different spanning trees does the graph on the right have?

(A) 2 (B) 3 (C) 4 (D) 5 (E) More than 5

1 circuit w/ 3 edges



lines

Suppose that after a storm an inspection needs to be made of the sewers along the streets in a small village to make sure local flooding is not due to clogging. The technique most likely to be useful in solving this problem is:

(A) finding an Euler circuit on a graph.
(B) applying the nearest-neighbor algorithm for finding a Hamiltonian circuit.
(C) applying Kruskal's algorithm for finding a minimum-cost spanning tree for a graph.
(D) applying the method of trees for the traveling salesman problem.
(E) None of these/need more information

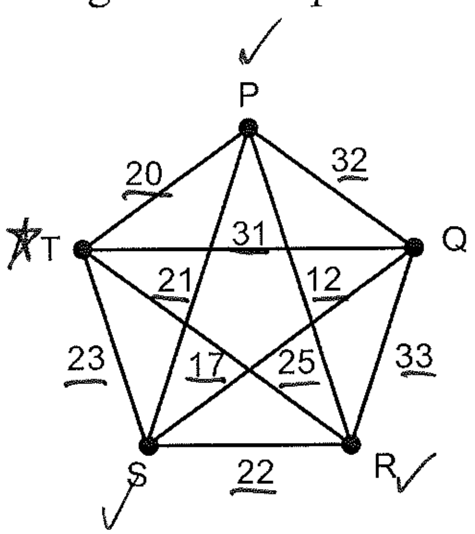
Which of the following statements are true about a spanning tree? Mark all true answers

(A) A spanning tree must contain all the edges of the original graph.
(B) A spanning tree must contain all of the vertices of the original graph.
(C) A spanning tree must contain a circuit
(D) A spanning tree must not contain a circuit
(E) None of these/need more information

Which, if any, of the statements below are true about a heuristic algorithm? Mark all true answers.

(A) An optimal result will be found
(B) An optimal result may be found
(C) An optimal result will not be found
(D) The brute force method is an example of a heuristic algorithm.
(E) The sorted edges method is an example of a heuristic algorithm.

The graph below shows the distance between houses in a rural neighborhood in yards. Use the nearest neighbor method to solve the traveling salesman problem for these houses starting at house T.

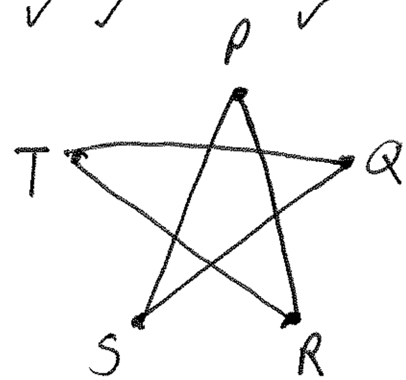


$T \xrightarrow{17} R \xrightarrow{12} P \xrightarrow{21} S \xrightarrow{25} Q \xrightarrow{31} T$

106 yds

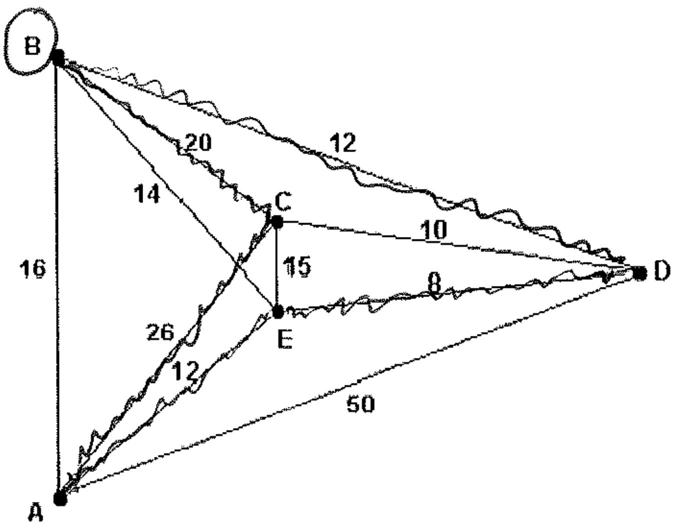
Use the sorted edges algorithm to solve the TSP for the graph above.

12 17 20 21 22 23 25 31 32 33



106 yds

For the graph below, find the Hamiltonian circuit obtained by using the nearest-neighbor algorithm, starting at B. What is the cost if the numbers shown represent the distance in miles?



BDEACB

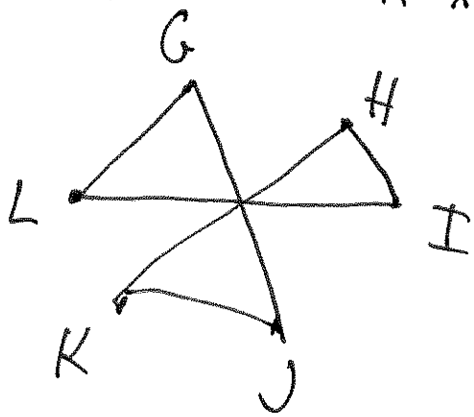
78 miles

The chart below shows the travel time between cities in hours. Use the sorted edges method to solve the TSP.

8 9 13 16 17 19

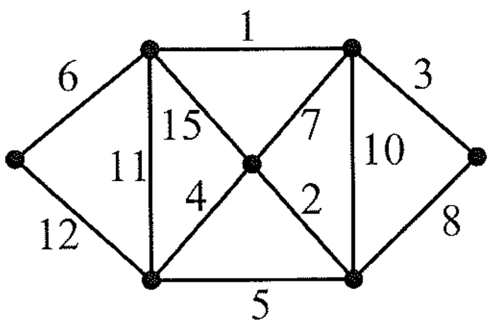
21 22 24 27 29 29 32 37 47

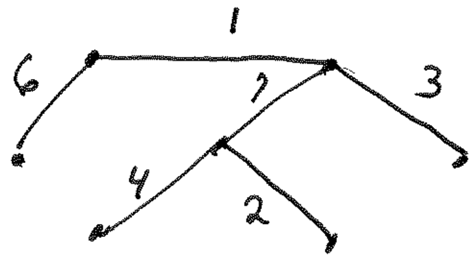
	G	H	I	J	K	L
G	0	16	21	13	32	9
H	16	0	19	29	37	29
I	21	19	0	47	27	8
J	13	29	47	0	24	17
K	32	37	27	24	0	22
L	9	29	8	17	22	0



110 hr

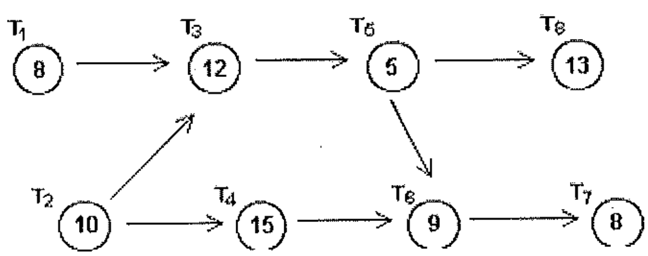
Apply Kruskal's algorithm to create a minimum cost spanning tree from the given graph. The edges show the time between vertices in seconds. What is the total time?





Cost 23 sec

What is the critical path for the digraph below? The time for each task is given in minutes.



$T_1\ T_3\ T_5\ T_8 = 38$ min

$T_1\ T_3\ T_5\ T_6\ T_7 = 42$ min

$T_2\ T_3\ T_5\ T_8 = 40$ min

$T_2\ T_3\ T_5\ T_6\ T_7 = 44$ min critical path

$T_2\ T_4\ T_6\ T_7 = 42$ min