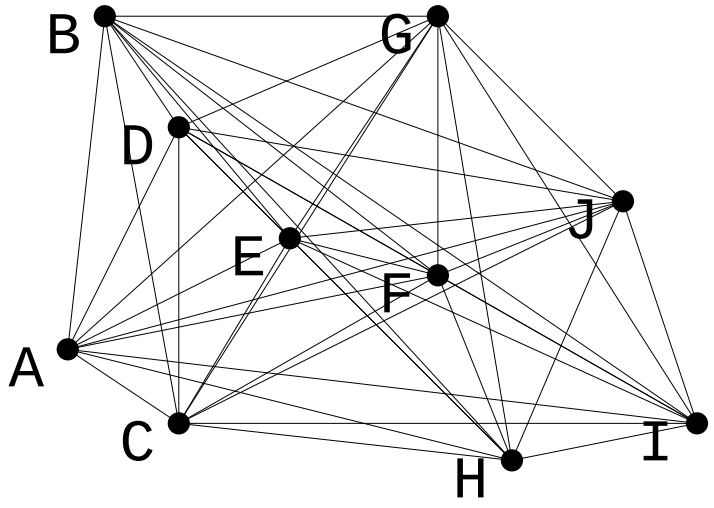
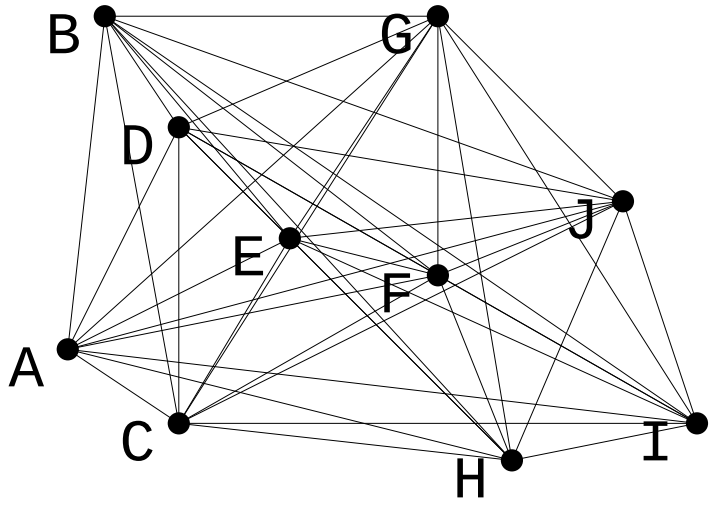


tsp 2-approx algorithm

* duplicate MST
* take 2-MST Eulerian tour
* shortcut tour (skip repeated nodes)
  => Hamiltonian cycle MSTHC

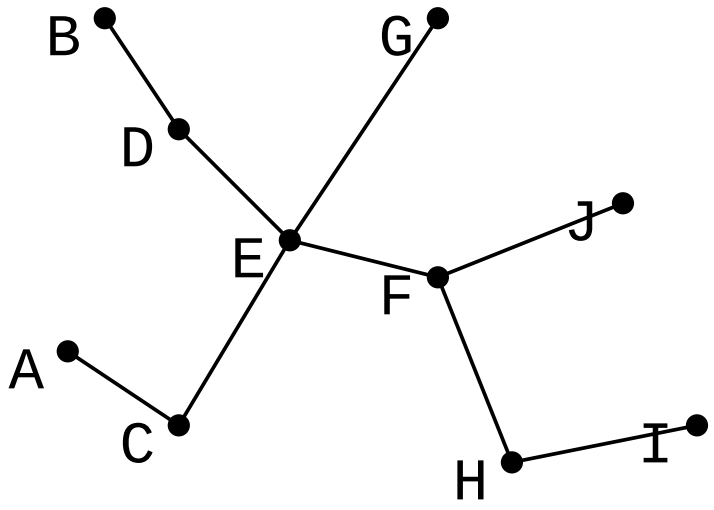
correctness

* wt(MSTHC) <= 2*wt(opt cycle)



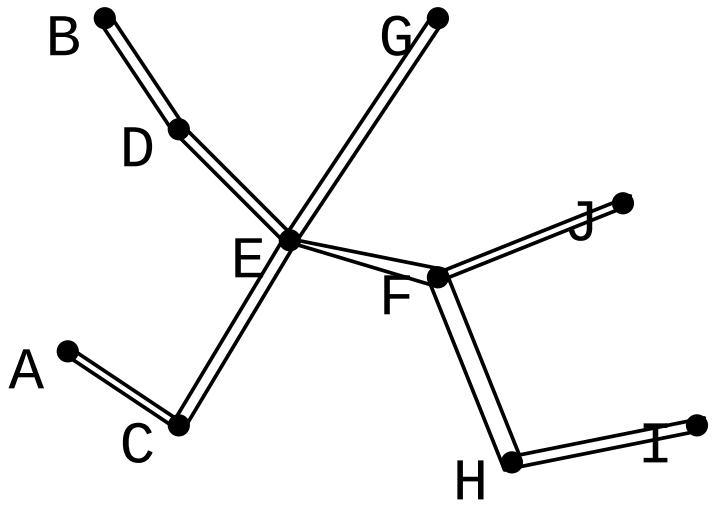
	A	B	C	D	E	F	G	H	I	J
A		5	2	4	4	6	7	7	9	9
B			6	2	4	6	5	9	11	8
C				4	3	4	7	5	8	7
D					2	4	4	7	9	7
E						2	4	5	7	5
F							4	3	4	3
G								7	7	4
H									3	4
I										4

input: complete weighted graph



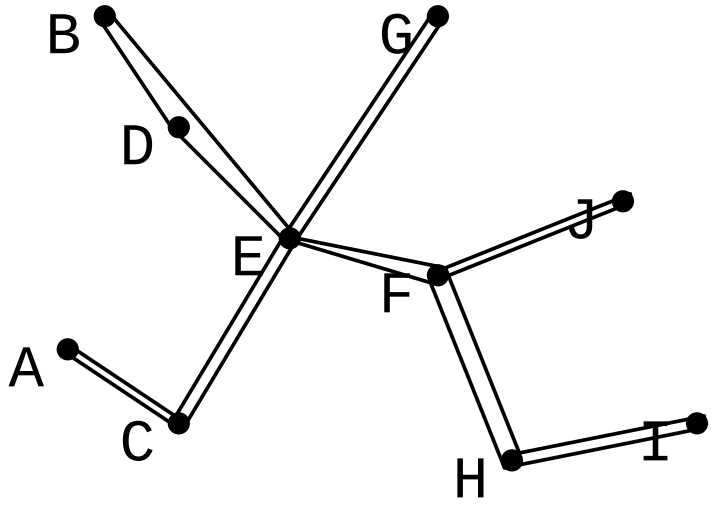
	A	B	C	D	E	F	G	H	I	J
A		5	2	4	4	6	7	7	9	9
B			6	2	4	6	5	9	11	8
C				4	3	4	7	5	8	7
D					2	4	4	7	9	7
E						2	4	5	7	5
F							4	3	4	3
G								7	7	4
H									3	4
I										4

find an MST weight 24



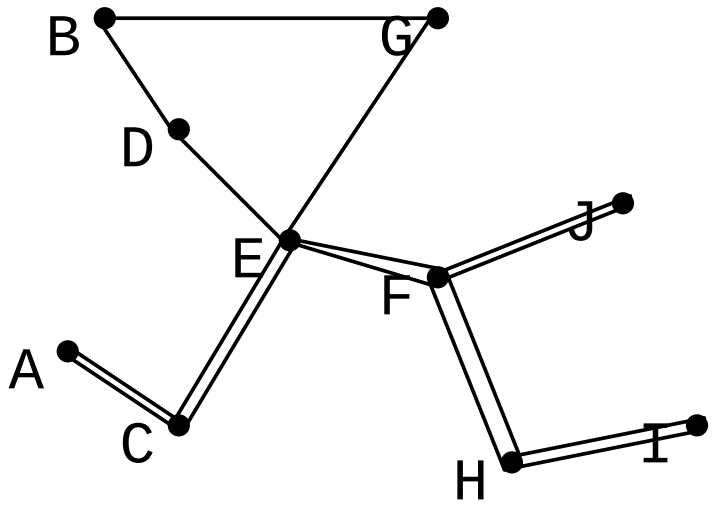
initial Hamiltonian tour:
  any duplicate-MST-Eulerian-tour

A C E D B D E G E F J F H I H F E C A



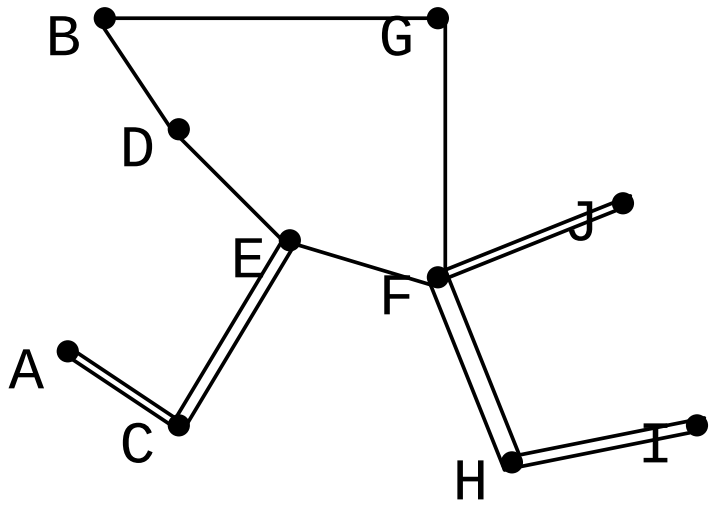
1st shortcut: skip repeated D

A C E D B E G E F J F H I H F E C A



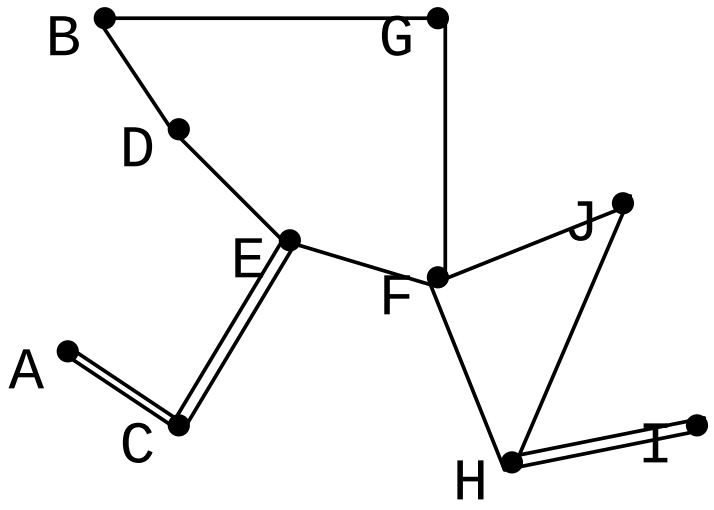
2nd shortcut: skip repeated E

A C E D B G E F J F H I H F E C A



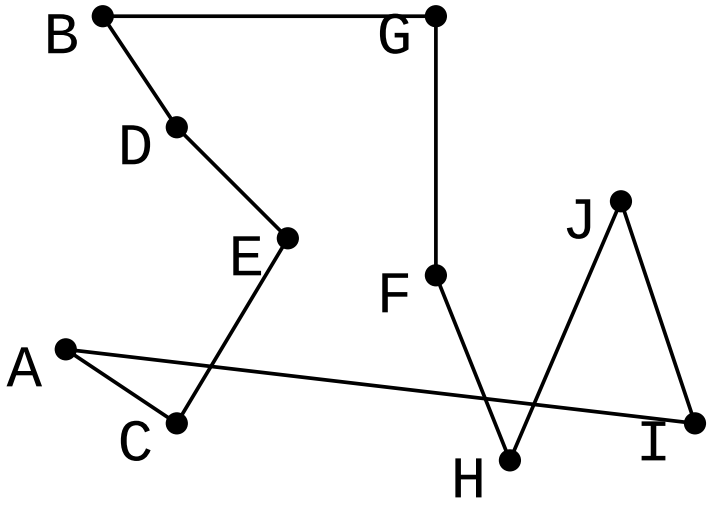
3rd shortcut: skip repeated E

A C E D B G F J F H I H F E C A



4th shortcut: skip repeated F

A C E D B G F J F H I H F E C A



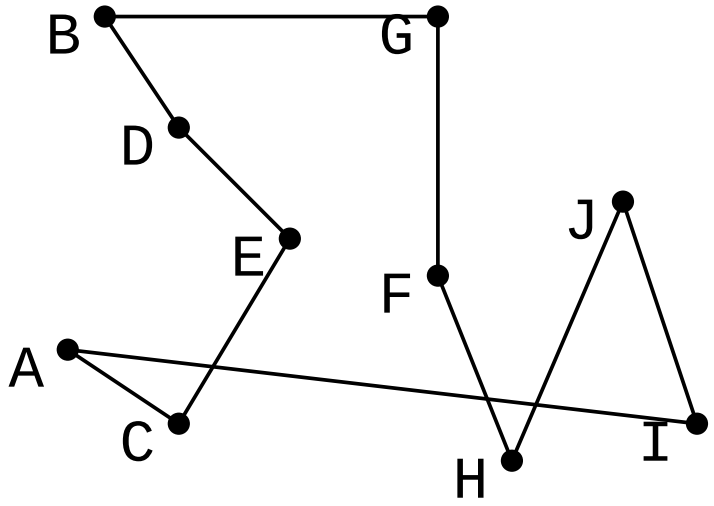
9th shortcut: skip repeated C

A C E D B G F H H I A

done! tour => cycle: no repeated node :)

this Hamiltonian cycle is

at-most-2 approx



(option: uncross each crossing)

uncross IA CE => IC AE

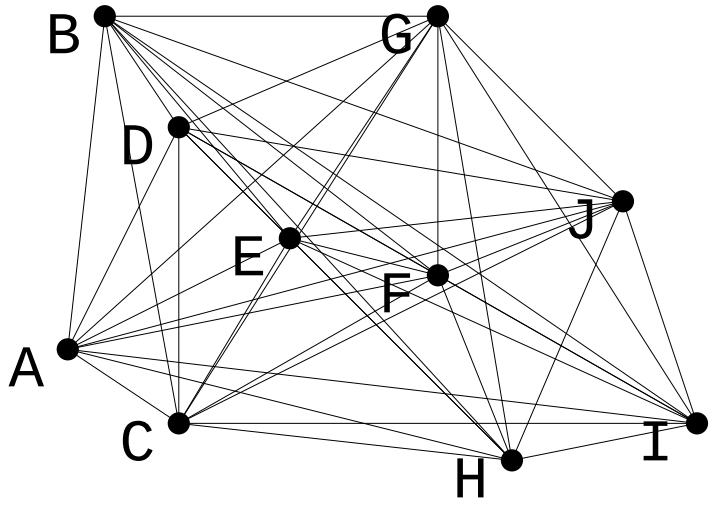
```
  A C E D B     G   F H   J I         A
  A   E D B     G   F H   J I   C     A

        FH IC => FI HC

  A   E D B     G   F H   J I   C     A
  A   E D B     G   F I   J H   C     A

        FI JH => FJ IH

  A   E D B     G   F I   J H   C     A
  A   E D B     G   F J   I H   C     A
```

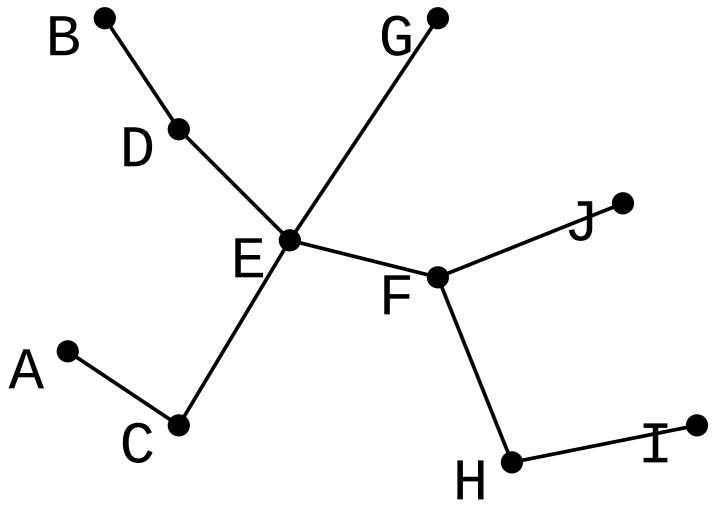


tsp 1.5-approx algorithm

* MST
* M: min matching on odd-degree-nodes of MST
* shortcut MST + M
  => Hamiltonian cycle MSTMHC

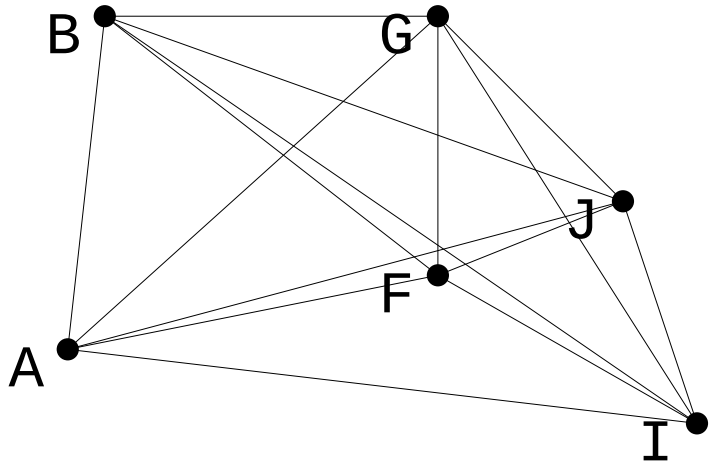
correctness

* wt(MSTMHC) <= 1.5*wt(opt cycle)



	A	B	C	D	E	F	G	H	I	J
A		5	2	4	4	6	7	7	9	9
B			6	2	4	6	5	9	11	8
C				4	3	4	7	5	8	7
D					2	4	4	7	9	7
E						2	4	5	7	5
F							4	3	4	3
G								7	7	4
H									3	4
I										4

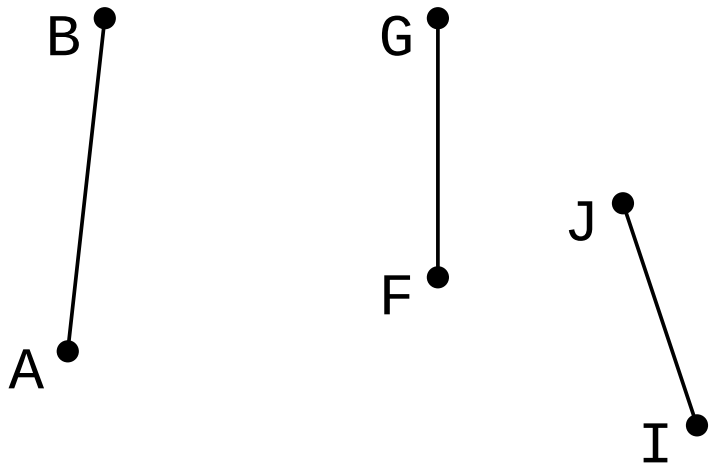
find an MST weight 24



complete subgraph induced by MST-odd-deg nodes:

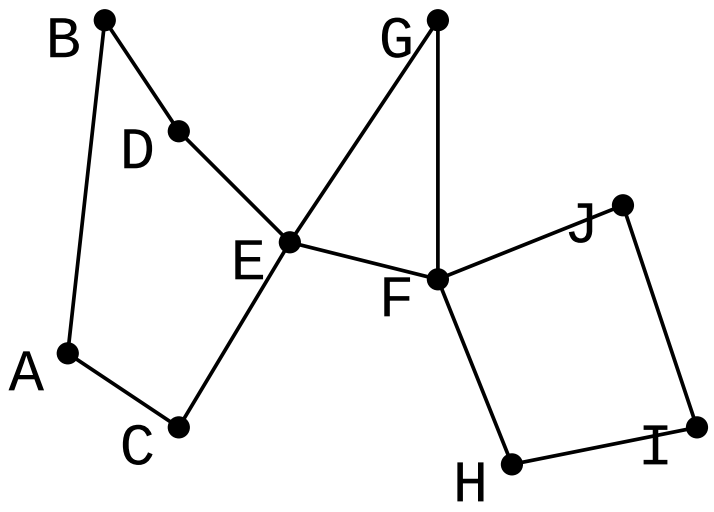
find min weight perfect matching M

	A	B	F	G	I	J
A		5	6	7	9	9
B			6	5	11	8
F				4	4	3
G					7	4
I						4



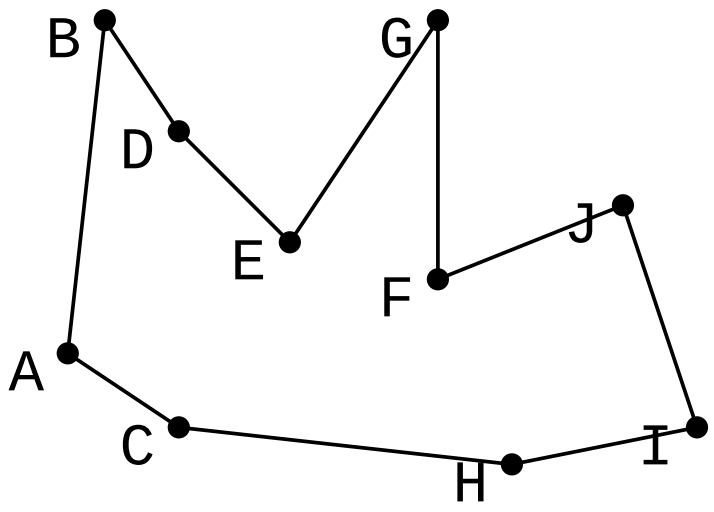
	A	B		F	G		I	J
A		5		6	7		9	9
B				6	5		11	8
F					4		4	3
G							7	4
I								4

min-weight perfect matching M wt 13



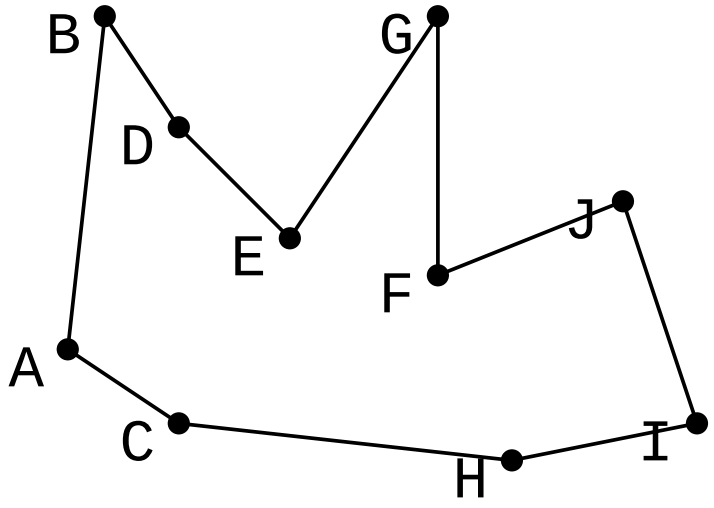
	A	B	C	D	E	F	G	H	I	J
A		5	2	4	4	6	7	7	9	9
B			6	2	4	6	5	9	11	8
C				4	3	4	7	5	8	7
D					2	4	4	7	9	7
E						2	4	5	7	5
F							4	3	4	3
G								7	7	4
H									3	4
I										4

MST + M wt 37



	A	B	C	D	E	F	G	H	I	J
A		5	2	4	4	6	7	7	9	9
B			6	2	4	6	5	9	11	8
C				4	3	4	7	5	8	7
D					2	4	4	7	9	7
E						2	4	5	7	5
F							4	3	4	3
G								7	7	4
H									3	4
I										4

shortcut MST+M 34

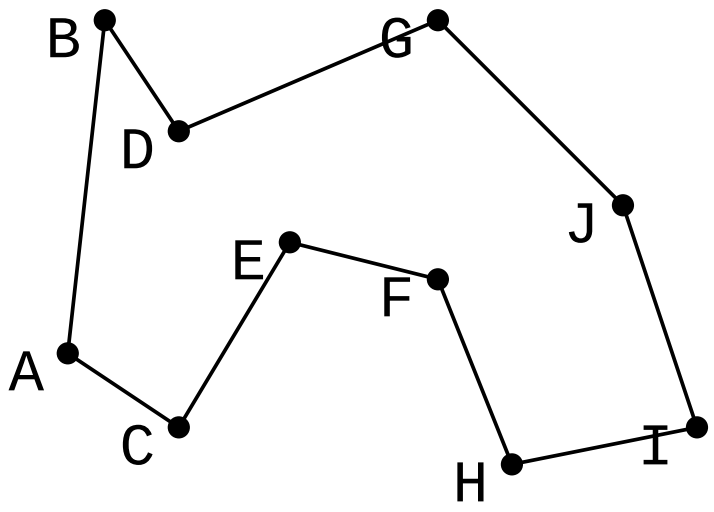


correctness

* wt(MST) <= wt(graph Hamiltonian cycle) why?
  : opt-cycle minus edge is spanning tree

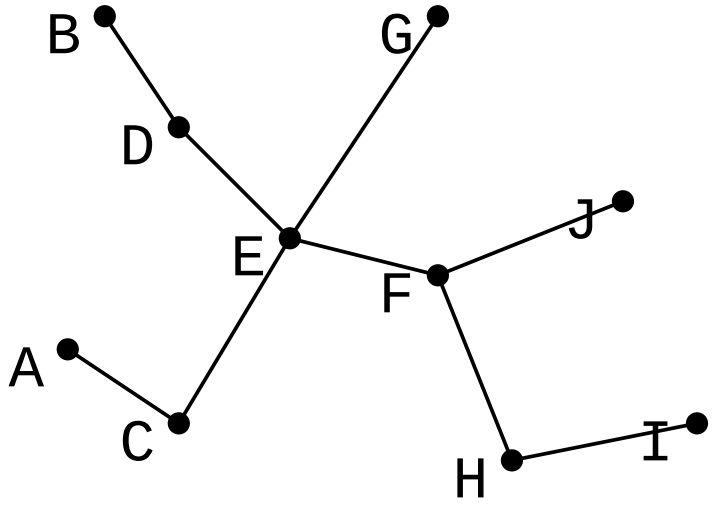
* wt(MST-odd-deg Ham-cycle) <= wt(graph Ham-cycle)
  : delete nodes, shortcut

* wt(MSTod min-wt perf-match) <= wt(MSTod Ham-cycle)/2
  : MSTod Ham-cycle partitions into M1 and M2
  : min of these has wt m0 <= wt(MSTod Ham-cycle)/2
  : wt(MSTod min-wt perf-match) <= m0



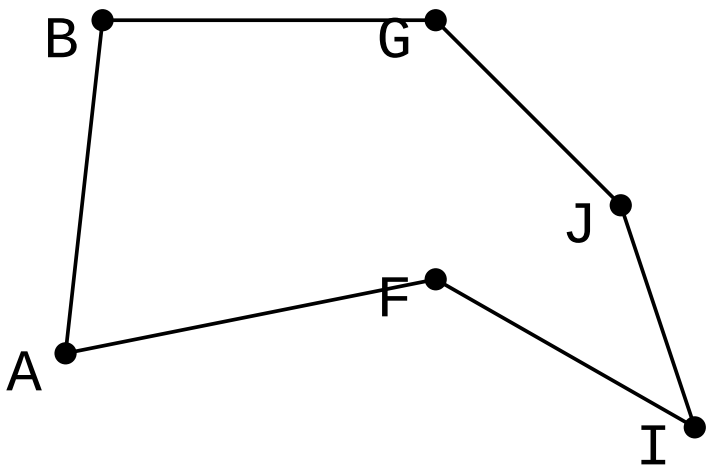
	A	B	C	D	E	F	G	H	I	J
A		5	2	4	4	6	7	7	9	9
B			6	2	4	6	5	9	11	8
C				4	3	4	7	5	8	7
D					2	4	4	7	9	7
E						2	4	5	7	5
F							4	3	4	3
G								7	7	4
H									3	4
I										4

optimal cycle (brute force) 32



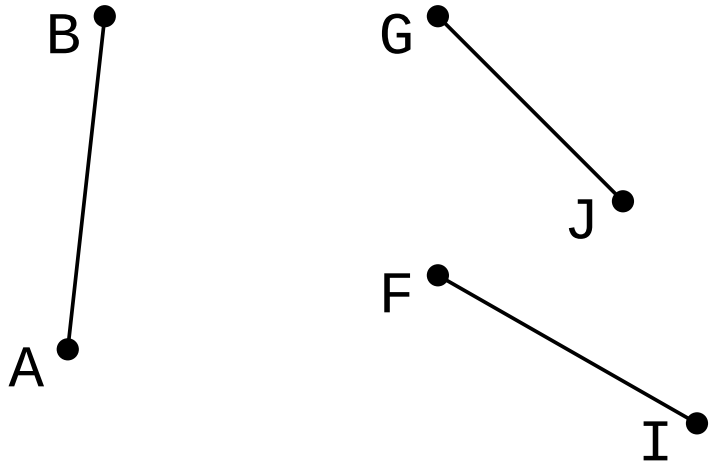
an MST weight 24

odd-degree nodes A B E G J I



	A	B		F	G		I	J
A		5		6	7		9	9
B				6	5		11	8
F					4		4	3
G							7	4
I								4

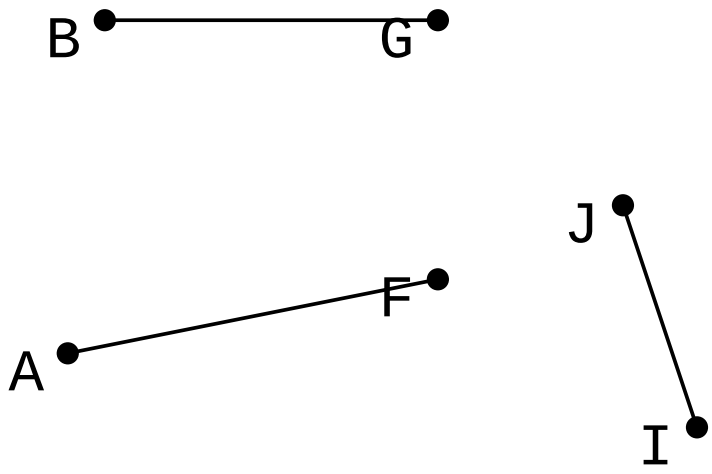
MST-odd-deg-nodes: opt-cycle wt 28



	A	B
A		5
B		
F		
G		
I		

	F	G	I	J
A	6	7	9	9
B	6	5	11	8
F		4	4	3
G			7	4
I				4

MST-odd-deg-nodes: opt-cycle, matching 1, wt 13



	A	B		F	G		I	J
A		5		6	7		9	9
B				6	5		11	8
F					4		4	3
G							7	4
I								4

MST-odd-deg-nodes: opt-cycle, matching 2, wt 15