The travelling salesman problem 5A

1 a

	A	*B*	*C*	*D*	*E*
A	-	7	10	9	5
B	7	-	3	11	12
C	10	3	-	8	12
D	9	11	8	-	4
E	5	12	12	4	-

AC – the shortest route is *ABC* length 10
AD – the shortest route is *AED* length 9
BD – the shortest route is *BCD* length 11
BE – the shortest route is *BAE* length 12

b

	A	*B*	*C*	*D*	*E*
A	-	5	12	7	9
B	5	-	7	2	4
C	12	7	-	5	7
D	7	2	5	-	2
E	9	4	7	2	-

AC – the shortest route is *ABDC* length 12
AE – the shortest route is *ABDE* length 9
BC – the shortest route is *BDC* length 7
BE – the shortest route is *BDE* length 4
CE – the shortest route is *CDE* length 7

c

	A	*B*	*C*	*D*	*E*	*F*
A	-	10	18	13	15	18
B	10	-	8	3	5	8
C	18	8	-	5	3	10
D	13	3	5	-	2	5
E	15	5	3	2	-	7
F	18	8	10	5	7	-

AC – the shortest route is *ABDEC* length 18
AF – the shortest route is *ABDF* length 18
BC – the shortest route is *BDEC* length 8
BE – the shortest route is *BDE* length 5
BF – the shortest route is *BDF* length 8
CD – the shortest route is *CED* length 5
CF – the shortest route is *CEDF* length 10
EF – the shortest route is *EDF* length 7

1 d

	A	*B*	*C*	*D*	*E*	*F*	*G*
A	-	10	9	10	17	20	20
B	10	-	3	20	11	10	20
C	9	3	-	19	8	13	23
D	10	20	19	-	27	20	10
E	17	11	8	27	-	8	18
F	20	10	13	20	8	-	10
G	20	20	23	10	18	10	-

AF – shortest route is *ABF* length 20
AG – the shortest route is *ADG* length 20
BE – the shortest route is *BCE* length 11
CF – the shortest route is *CBF* length 13
CG – the shortest route is *CBFG* length 23
DE – the shortest route is *DACE* length 27

2 a In the classical problem, each vertex must be visited exactly once. In the practical problem, each vertex must be visited at least once.

b i It is helpful to find the shortest path from *A* to *D*, as this is the most distant node from *A*. If possible, we should also avoid edges of high weight (*AE* and *CD*). Keeping that in mind, by inspection we find that the solution is *ABECEDEBA* (or *ABEDECEBA*). Weight of the tour is $2+1+3+3+4+4+1+2=20$

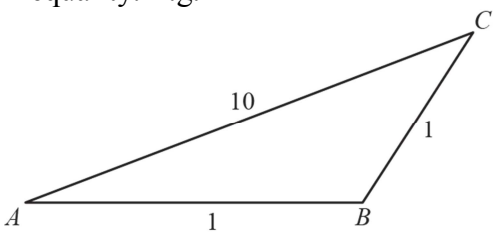
ii *ABCDEA* and *AEDCBA* are in fact the only tours starting at *A* which visit every vertex exactly once. Hence, either of them provides solution to the classical problem. Total weight is $2+3+9+4+8=26$

3 a The solution must begin and end with edge *PQ*. By inspection we find *PQRTSUQP* (or *PQUSTRQP*) as a solution. Observe that *QRTSUQ* contains 5 edges of least weight in the graph. As the solution requires at least 7 edges (2 of them being *PQ),* we conclude it is optimal. Total weight $7+3+5+6+8+3+7=39$

b Regardless of the starting vertex, we need to include edge *PQ* twice. The remaining problem is to find the solution to the practical problem for subgraph consisting of vertices *Q,R,S,T,U* and edges between them. It must contain at least 5 edges. As discussed in part **a**, optimal solution is provided by 5 edges of least weight. We can form a cycle starting at any vertex out of them, e.g. *QRTSUQ* starting in *Q* or *RTSUQR* starting in *R.* By adding *PQ* in appropriate position we can construct the solution or total weight 39 starting at any point, e.g. *QPQRTSUQ* starting in *Q* or *RTSUQPQR* starting in *R.*

c As *PQ* is the only edge connecting *P* to the rest of the graph, we must visit *Q* at least twice.

4 The simplest example is a graph with 3 vertices where the weights of edges do not satisfy triangle inequality. E.g.



Challenge

1 In a complete graph any two vertices are connected. Hence, after fixing the start vertex, you can visit all other vertices in arbitrary order. The tour of that type which has minimum total weight provides the solution to the classic problem.

For example consider K_4 with vertices *A,B,C,D* and take A as a start vertex. Since there is an edge between any two nodes, *ABCDA, ABDCA, ACBDA, ACDBA, ADBCA* and *ADCBA* are all possible tours. The one of minimum weight provides a solution to classical problems. Similarly, for any other vertex.

2 Count distinct Hamiltonian cycles in a complete graph K_n To this end, fix the start vertex and consider all Hamiltonian cycles starting from there. There are $(n-1)!$ of them, as we can visit all other vertices in arbitrary order $(n-1)!$ is the number of permutations of $n-1$ elements). We need to account for the fact that any cycle can be traversed both ways, so effectively there are $\frac{(n-1)!}{2}$ distinct Hamiltonian cycles.

Selecting the minimal of $\frac{(n-1)!}{2}$ numbers requires $\frac{(n-1)!}{2}-1$ comparisons. Hence, the order of algorithm is $\frac{(n-1)!}{2}$ As factorial grows very rapidly, it is practically impossible to use the algorithm for larger graphs.