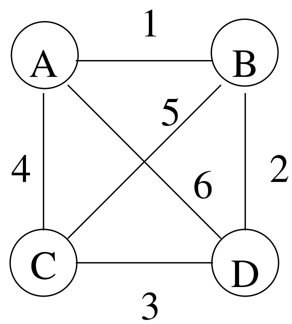
Ally Smith

**TSP Worksheet**




1. Show all possible TSP tours in the graph and compute their cost; for example, one TSP tour is $A-B-C-D-A$ and its cost is $1+5+3+6=15$.

   $A-B-C-D-A$, $1+5+3+6=15$

   $A-B-D-C-A$, $1+2+3+4=10$

   $A-C-B-D-A$, $4+5+2+6=17$

   $A-C-D-B-A$, $4+3+2+1=10$

   $A-D-B-C-A$, $6+2+5+4=17$

   $A-D-C-B-A$, $6+3+5+1=15$

   $B-A-C-D-B$, $1+4+3+2=10$

   $B-A-D-C-B$, $1+6+3+5=15$

   $B-C-A-D-B$, $5+4+6+2=17$

   $B-C-D-A-B$, $5+3+6+1=15$

   $B-D-A-C-B$, $2+6+4+5=17$

   $B-D-C-A-B$, $2+3+4+1=10$

   $C-A-B-D-C$, $4+1+2+3=10$

   $C-A-D-B-C$, $4+6+2+5=17$

   $C-B-A-D-C$, $5+1+6+3=15$

   $C-B-D-A-C$, $5+2+6+4=17$

   $C-D-A-B-C$, $3+6+1+5=15$

   $C-D-B-A-C$, $3+2+1+4=10$

   $D-A-B-C-D$, $6+1+5+3=15$

   $D-A-C-B-D$, $6+4+5+2=17$

   $D-B-A-C-D$, $2+1+4+3=10$

   $D-B-C-A-D$, $2+5+4+6=17$

   $D-C-A-B-D$, $3+4+1+2=10$

$D - C - B - A - D$, $3 + 5 + 1 + 6 = 15$

2. How many distinct tours are there when you account for the same tour being counted multiple times?

   3 distinct tours

**Counterexample Worksheet**

The discovery of an algorithm often begins with a sudden insight into the problem. Sometimes (unfortunately), an idea that seems very intuitive at first glance, turns out not to be correct on further thought. Figuring out that something is not correct by finding counterexamples is a useful skill. Among other things, it deepens your understanding of the problem.

Find counterexamples for the following propositions:

1. **Proposition**: $a + b > \min(a, b)$

   $a = -1$, $b = -2$

   $a + b = -3$

   $min(a, b) = -2$

   $-3 < -2$

2. **Proposition**: the shortest route in a road network between two points is one with the fewest turns.

   It is possible for a shorter road to involve more turns. For example, if a road is made up of three segments each a quarter-mile long, and there is another route that is two half-mile sections, then the shortest route would have two turns, but a longer route would exist with fewer turns.

3. **Proposition**: being up by a queen in a game of chess guarantees a win!

   It is possible to be up by a queen and still lose in chess. For example, your opponent could be missing their queen, but instead use a rook and a bishop to corner your king and put you in checkmate.

## Insertion-Sort/Execution-Counter Worksheet

Assume array $A$ is indexed from 1 to $n$.

```
INSERTION_SORT(A, n)
1. for j ← 2 to n do
2.     key ← A[j]
3.     i ← j − 1;
4.     while i > 0 and A[i] > key do
5.         A[i + 1] ← A[i]
6.         i ← i − 1
7.     A[i + 1] ← key
```

Instance 1 : [4, 3, 2, 1]
Instance 2 : [1, 4, 2, 3]
Instance 3 : [5, 4, 3, 2, 1]
Instance 4 : [1, 2, 3, 4]

| | # Times Executed | | | |
|---|---|---|---|---|
| Line No | Instance 1 | Instance 2 | Instance 3 | Instance 4 |
| L1 | 3 | 3 | 4 | 3 |
| L2 | 3 | 3 | 4 | 3 |
| L3 | 3 | 3 | 4 | 3 |
| L4 | 8 | 7 | 16 | 3 |
| L5 | 6 | 2 | 12 | 0 |
| L6 | 6 | 2 | 12 | 0 |
| L7 | 3 | 3 | 4 | 3 |
| Total | 32 | 23 | 56 | 15 |

List any observations.

In instance 1, the list is in reverse order and has $2n^2$ operations, while the instance that I created was the same size and already sorted, and had $n^2 - 1$ steps (according to my calculations).

More generally speaking, the closer to being sorted an input was, the fewer instructions it took, as long as the size was held constant. However, even with a small increase in the length of the input the number of times each line was executed jumped dramatically.