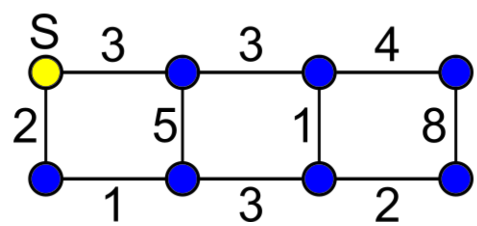
CEMC at Home

Grade 11/12 - Wednesday, March 25, 2020
Byber Path

You drive for a delivery service called Byber. You start at location S and you have to drop off a package at each of the seven other locations, shown as circles. The locations are joined by roads shown as lines. You cannot visit any location more than once on your route. You can finish at any location that you wish. The number beside a line is the toll you need to pay for taking the corresponding road.



What is the least total amount you need to pay in order to drop off all seven packages?

You may end up with the correct answer for this problem without being completely convinced that your answer is indeed correct! Think about what it would take to completely justify the validity of your answer. You would need to show that there is a route that will result in exactly your amount, and explain why every other route would cost at least as much as yours.

More Info:

Check the CEMC at Home webpage on Wednesday, April 1 for the correct answer with justification.

This problem almost ended up on the 2019 Beaver Computing Challenge (BCC) which is a problem solving contest with a focus on computational and logical thinking.

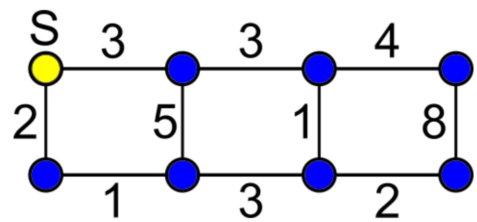
While only officially open to students in Grades 5 to 10, students in Grades 11 and 12 can have fun and learn something by trying the BCC problems. You can find more problems like this on past BCC contests.

CEMC at Home

Grade 11/12 - Wednesday, March 25, 2020
Byber Path - Solution

Problem

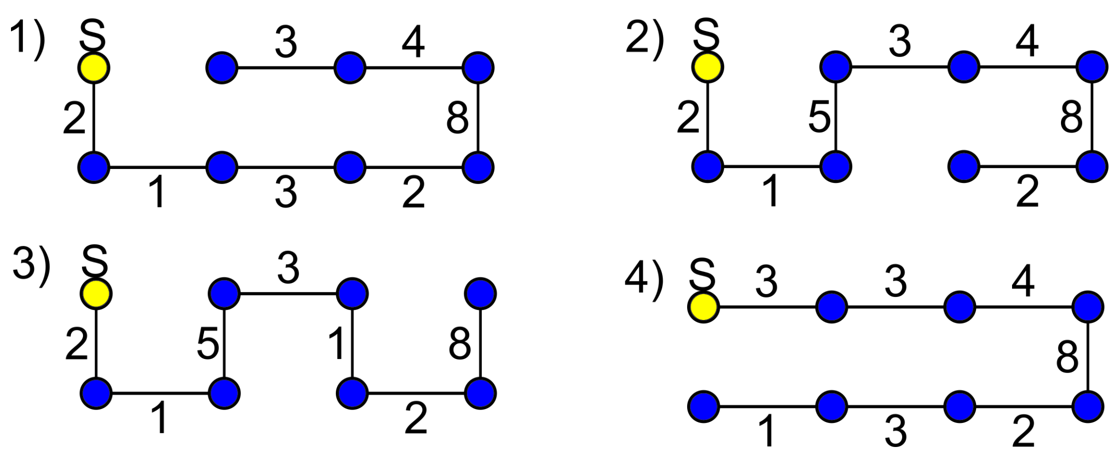
You drive for a delivery service called Byber. You start at location S and you have to drop off a package at each of the seven other locations, shown as circles. The locations are joined by roads shown as lines. You cannot visit any location more than once on your route. You can finish at any location that you wish. The number beside a line is the toll you need to pay for taking the corresponding road.



What is the least total amount you need to pay in order to drop off all seven packages?

Solution

There are only four possible paths that start at S and visit each of the locations exactly once:



To justify this, notice that we have two choices for where to travel from S: down or right.

- If we travel right first, then the rest of our path is determined. If we move down before reaching the right-most location in the top row, then we would need to backtrack to reach all of the bottom locations. This would mean revisiting at least one location on the way. Therefore, we must travel right until we reach the right-most location in the top row, and then must follow path 4) for the remainder.

- If we travel down first, then we must travel right next, but then there we again have two options: up or right.
  - If we travel right, then the rest of our path is determined. We cannot move up next, as then we cannot reach the right-most locations without backtracking, and so we must move right to get to the right-most bottom location. From here we would have to follow path 1) for the remainder.
  - If we travel up, then we must travel right next. From here we have two choices: down or right.
    * If we travel down, then the rest of our path is determined. We must follow path 3).
    * If we travel right, then the rest of our path is determined. We must follow path 2).

These four possible paths have total paid amounts of:

1) $2 + 1 + 3 + 2 + 8 + 4 + 3 = 23$
2) $2 + 1 + 5 + 3 + 4 + 8 + 2 = 25$
3) $2 + 1 + 5 + 3 + 1 + 2 + 8 = 22$
4) $3 + 3 + 4 + 8 + 2 + 3 + 1 = 24$

(where the sum is shown starting at S and moving through the path).

The least total paid amount of these four paths is 22.

Connections to Computer Science

For each Beaver Computing Challenge problem, we include a short description of its connections to computer science. The italicized keywords emphasize terminology that can be used to search online, if you are interested in learning more.

In this particular problem, the locations and roads can be modelled by a *graph*. Locations are the *vertices* and roads are the *edges* of the graph. To use a computer to solve a problem like this, we need to

- figure out how to represent the graph, and
- discover and implement an *algorithm* to produce the final answer.

In a *programming language*, different *data structures* exist or can be built to represent virtually anything we can image. Amazingly, everything is ultimately modelled by a long *binary* sequence of 0s and 1s.

Different computer algorithms are used for finding the best or the worst path through a graph. In this problem, one of the restrictions is to find a path which visits all the vertices exactly once. This is called a *Hamiltonian path*. Problems involving Hamiltonian paths are well-known and considered very difficult. They are closely related to what is known as the famous *Travelling salesman problem*. You can read more about this problem here.