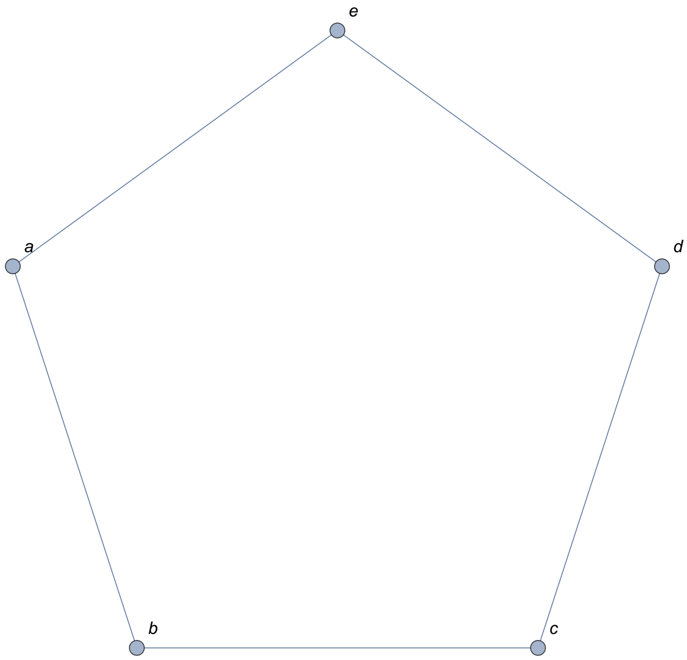
Nicole is running on a graph given by ‘edge’. She is now at vertex ‘x’ and needs to finish at home vertex ‘y’. Nicole wants to exercise as much as possible while she only has ‘n’ chocolate bars, each of which provides the energy for Nicole to run through a single edge. Yuan Yuan finds a route for Nicole to run.

For example, let 'edge' be given by its edges as follows.

```
G = Graph[{a ⟷ b, b ⟷ c, c ⟷ d, d ⟷ e, e ⟷ a}, VertexLabels → All]
```

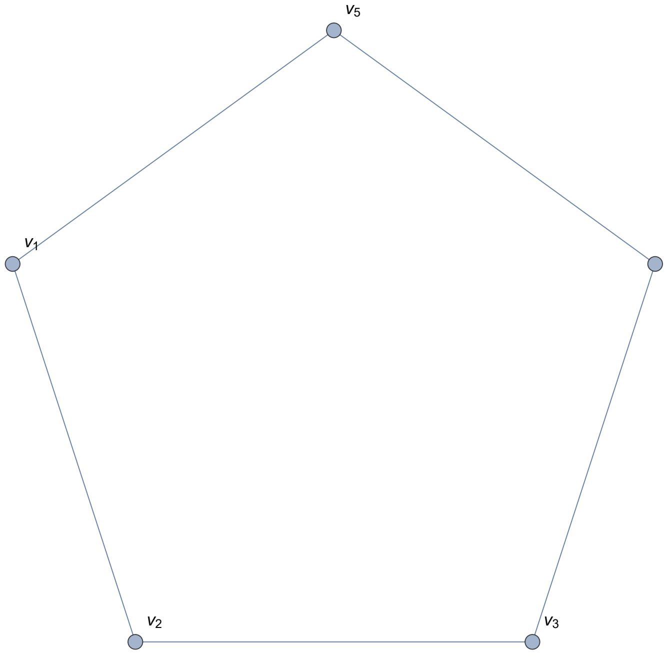


```
edge = {a ⟷ b, b ⟷ c, c ⟷ d, d ⟷ e, e ⟷ a};
```

```
AdjacencyMatrix[edge]
```

```
SparseArray[ Specified elements: 10
             Dimensions: {5, 5} ]
```

```
AdjacencyGraph[AdjacencyMatrix[edge],
 VertexLabels → Table[i → v_i, {i, First[Dimensions[AdjacencyMatrix[edge]]]}]]
```



```
Dimensions[AdjacencyMatrix[edge]]
```

```
{5, 5}
```

```
walk[edge_, x_, y_, n_] := MatrixPower[AdjacencyMatrix[edge], n]〚x, y〛
```

```
walk[edge, 1, 3, 4]
```

```
4
```

```
path[_, x_, x_, 0] := ToString[v_x, StandardForm];
path[edges_, x_, y_, n_]  := Module[
  {M = Normal[AdjacencyMatrix[edges]], z},
  z = RandomChoice[Level[Intersection[
      Position[MatrixPower[M, n - 1][[x]], w_  /; w ≠ 0],
      Position[M[[y]], w_ /; w ≠ 0]
     ], {2}]];
  path[edges, x, z, n - 1] <> ToString[v_z, StandardForm]
 ]
```

```
path[edge, 1, 3, 4]
```

```
v_1v_1v_2v_1v_2
```