# 5CCS2FC2: Foundations of Computing II

Tutorial Sheet 4

4.1 The problem **VERTEX-COVER** takes as input a graph $G = (V, E)$ and integer $k > 0$ and returns `True` if there is a set $C \subseteq V$ of size $k$ such that every edge in $E$ connects to some vertex in $C$. (The set $C$ is called a *vertex-cover*. See `https://en.wikipedia.org/wiki/Vertex_cover`)

Consider the following polynomial reduction from **SAT** to **VERTEX-COVER**:

> **Step 1)** Let $F$ be a formula in conjunctive normal form (CNF) with three literals in each clause,
>
> **Step 2)** Construct a graph $G_F = (V, E)$, where
>
> $$V = \{L^i \ : \ L \text{ is a literal belonding to the } i\text{th clause}\}$$
>
> and
>
> $$(L_1^i, L_2^j) \in E \qquad \Longleftrightarrow \qquad i = j \quad \text{or} \quad L_1 \equiv \neg L_2$$
>
> That is to say that two literals are connected with an edge if they appear in the same clause, or if they are contradictory (*e.g.*, $P$ and $\neg P$).
>
> **Step 3)** Return the pair $\langle G_F, k \rangle$, where $k$ is twice the number of clauses in $F$, with the property that
>
> $$F \in \textbf{SAT} \qquad \Longleftrightarrow \qquad \langle G_F, k \rangle \in \textbf{VERTEX-COVER}.$$

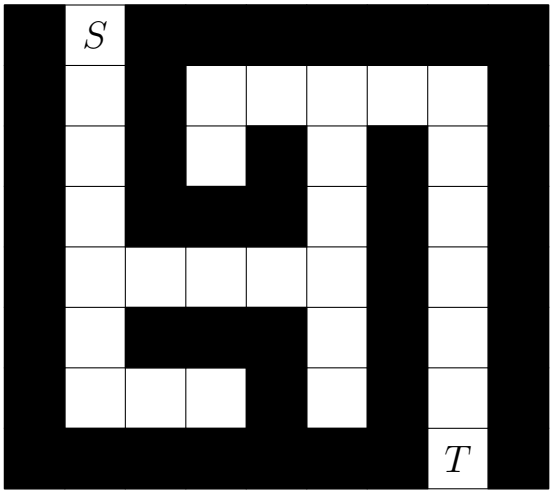
(i) Construct the graph $G_F$ for the following formula

$$F \ = \ (P \vee Q \vee R) \wedge (\neg P \vee \neg Q \vee \neg R) \wedge (\neg P \vee Q \vee \neg R) \wedge (P \vee \neg Q \vee R)$$

(ii) Find a vertex cover of size $k = 8$.

(iii) What can you say about the vertices that do not belong to the vertex cover?

4.2 Consider the following maze:




(i) Convert the maze into a graph by replacing the cells with vertices and the possible paths with edges.
*Hint: Try to minimise the number of vertices required to fully describe the structure of the maze.*

(ii) Find a route through the maze from $S$ to $T$ using Breadth-First-Search (BFS)

(iii) Find a route through the maze from $S$ to $T$ using Depth-First-Search (DFS).

(iv) How does the order in which the vertices are added to the queue/stack affect each of the search algorithms?

**Remember to complete the Week 4 Feedback for your TAs:**

`https://keats.kcl.ac.uk/mod/feedback/view.php?id=2054168`