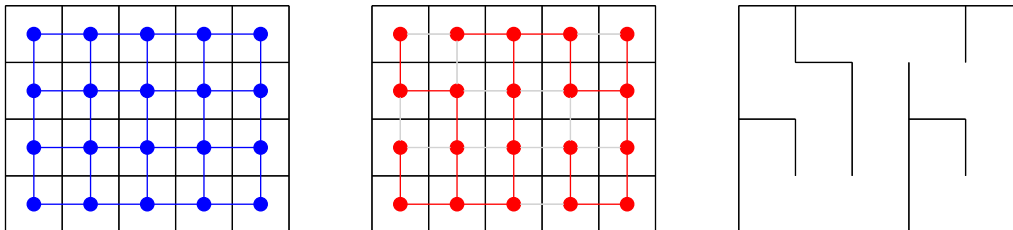


Objectifs et travail à effectuer

Il existe de nombreuses approches pour générer aléatoirement des labyrinthes. L'une des plus simples repose sur l'exploration en profondeur d'un graphe. Plus spécifiquement on considère une grille de n rangées par m colonnes à laquelle on associe un graphe dont les sommets correspondent aux cellules de la grille et où deux sommets sont reliés par une arête si les cellules associées sont contigües (horizontalement ou verticalement). La situation est illustrée ci-dessous à gauche pour une grille de 4 rangées par 5 colonnes.



On choisit ensuite un sommet du graphe comme point de départ d'une exploration en profondeur dans laquelle les voisins de chaque sommet sont traités dans un ordre aléatoire. On obtient alors un arbre recouvrant comme illustré ci-dessus au centre. Pour chaque arête de cet arbre on supprime le segment correspondant de la grille, autorisant ainsi le passage entre les deux cellules associées aux extrémités de l'arête. Comme résultat on obtient un labyrinthe, généré aléatoirement, comme illustré ci-dessus à droite.

Dans ce travail pratique vous devez programmer la construction et la gestion du graphe associé à une grille ainsi que son exploration en profondeur en complétant l'implémentation des classes `GridGraph` et `DFSMazeGenerator` dans les sources fournies.

Modalités et délais

- ▷ Le travail de programmation est à effectuer par groupe de deux, en Java, version 17.
- ▷ L'archive contenant les sources du projet est disponible sur le site Cyberlearn du cours.
- ▷ Vous devez rendre une archive (au format `zip`) contenant toutes les sources de votre projet complété et vous prêterez une attention toute particulière aux commentaires de vos implémentations des classes `GridGraph` et `DFSMazeGenerator`.
- ▷ Vous devez rendre votre travail sur Cyberlearn au plus tard le **dimanche 26 mars 2023** (avant minuit).