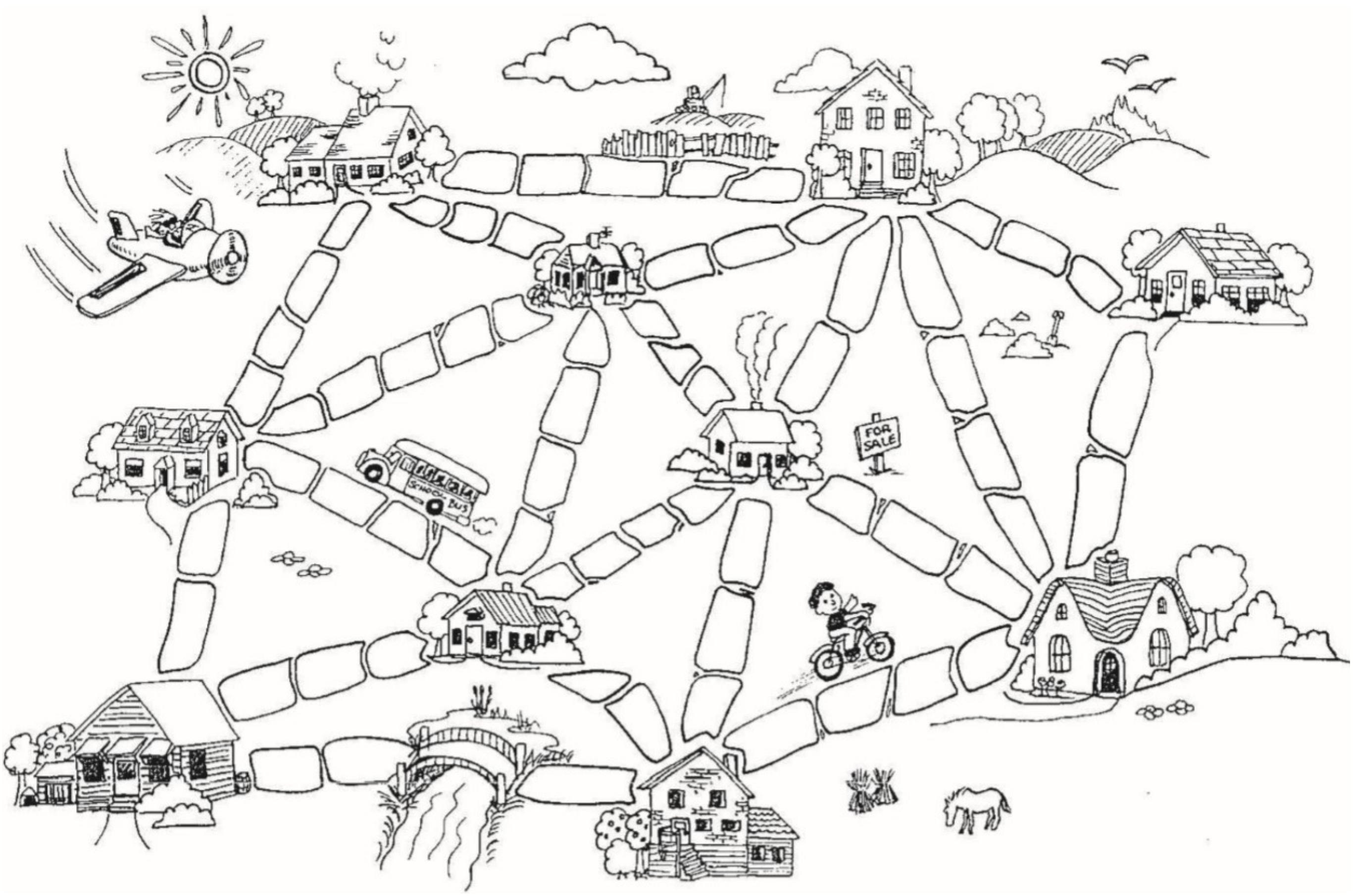
<u>Activité débranchée :</u> Arbre couvrant

Dans la ville « Gadoue », il est très difficile de circuler après de fortes pluies car le sol devient boueux faute de rues pavées. Le maire décide donc de paver certaines rues, mais en respectant deux conditions :

1) Paver suffisamment de rues pour que n'importe quel habitant puisse se rendre de sa maison à n'importe quelle autre maison en empruntant les rues pavées.

2) Dépenser le moins d'argent possible pour paver ces rues.

L'agencement de la ville est représenté ci-dessous. Le nombre de pavés entre chaque maison représente la dépense à engager pour paver la route.



1) a) Trouve le meilleur chemin pour relier toutes les maisons mais utilise le moins de jetons (pavés) possible.

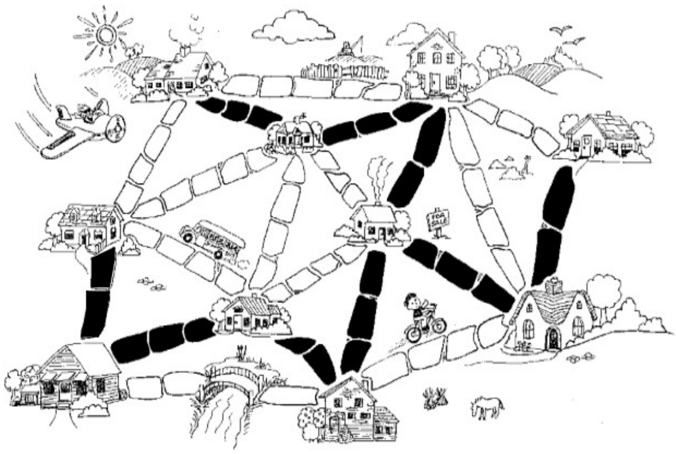
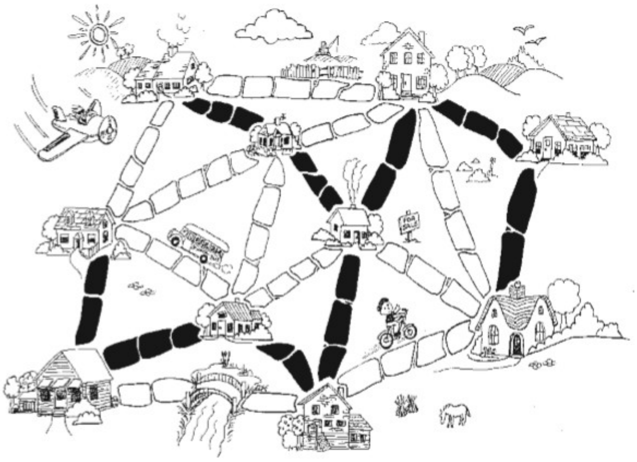
b) Combien de pavés sont nécessaires pour ce chemin ?

c) Aurait-on pu faire mieux ? Justifie !

2) Propose un algorithme permettant de paver n'importe quelle ville en disposant d'un plan semblable et en respectant les deux conditions de l'énoncé.

Correction :

1) a) Il existe plusieurs chemins optimaux (deux exemples ici) :

b) Chaque chemin optimal nécessite 23 pavés.

c) Il y avait 10 maisons à relier entre elles.

- On choisit une première maison. Cette dernière est forcément reliée par 1 rue à une autre maison : 2 maisons → 1 rue
- Parmi ces 2 maisons, au moins 1 doit être reliée à une 3ème maison : 3 maisons → 2 rues
- …
- 10 maisons → 9 rues

Il faut donc sélectionner 10 rues. Les 10 rues les plus courtes : 4 rues de 2 pavés et 5 rues de 3 pavés ce qui fait au minimum 23 pavés. Nous avons donc bien trouvé un chemin optimal.

2) **Rues_sélectionnées** = vide

Liste_maisons = Choisir une maison quelconque pour commencer

Répéter jusqu'à ce que Liste_maisons contiennent toutes les maisons de la ville :

- ***Rues_candidates*** *= Sélectionner toutes les rues dont une extrémité et une seule est une maison de* ***Liste_maisons***   - ***Rue_à_ajouter*** *= Rue la plus courte parmi* ***Rues_candidates***   *- Ajouter* ***Rue_à_ajouter*** *à* ***Rues_sélectionnées***   *- Ajouter à* ***Liste_maisons*** *la maison (qui n'est pas dans Liste_maisons) située à l'autre extrémité de* ***Rue_à_ajouter***

Pour aller plus loin :

1) Dans le cas où on a n maison dans une ville, combien de rues sont nécessaires pour toutes les relier ?

2) Trouver une disposition simples des maisons dans laquelle les $n-1$ rues ne sont pas les $n-1$ plus courte.

3) Modéliser à l'aide d'un graphe.

4) <u>*Utilisations concrètes :*</u>

Suppose qu'il faille mettre en place la distribution de l'électricité, du gaz ou de l'eau dans une nouvelle commune. Il est nécessaire de créer un réseau de câbles ou de tuyaux pour relier toutes les maisons à la compagnie de distribution. Chaque maison doit être reliée au réseau en un point et la route choisie pour atteindre la maison importe peu, du moment qu'elle existe.

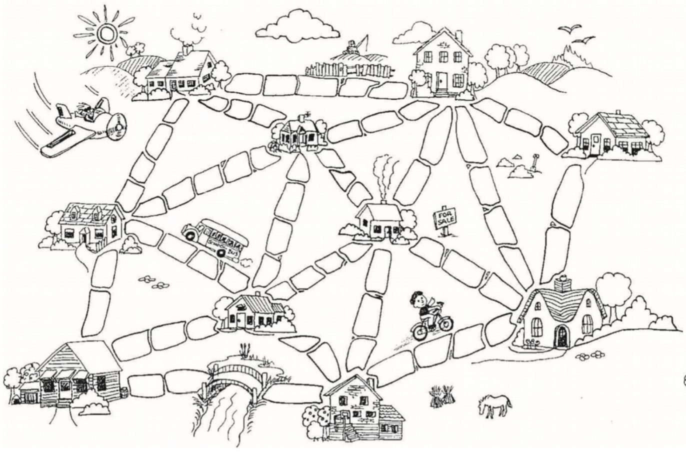
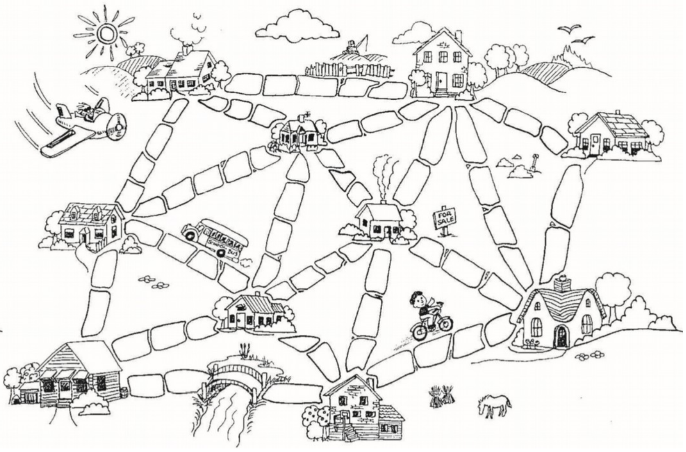
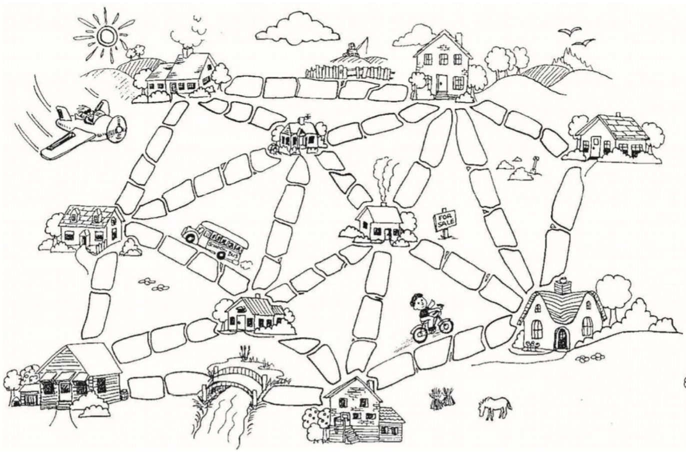
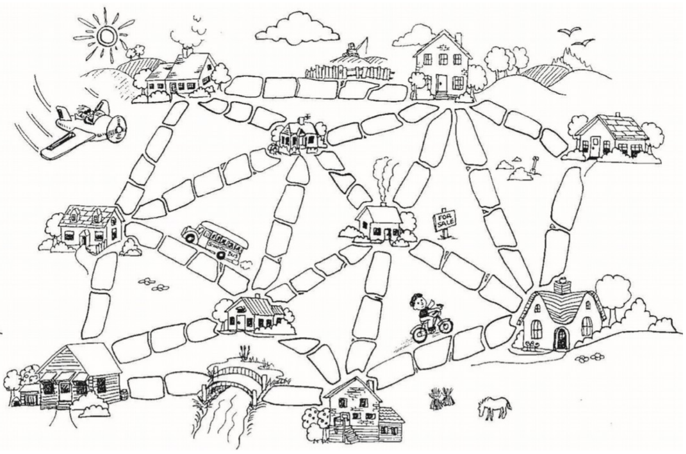
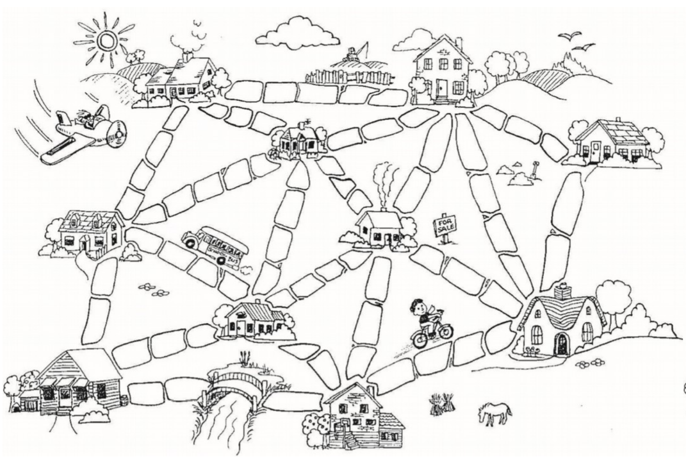
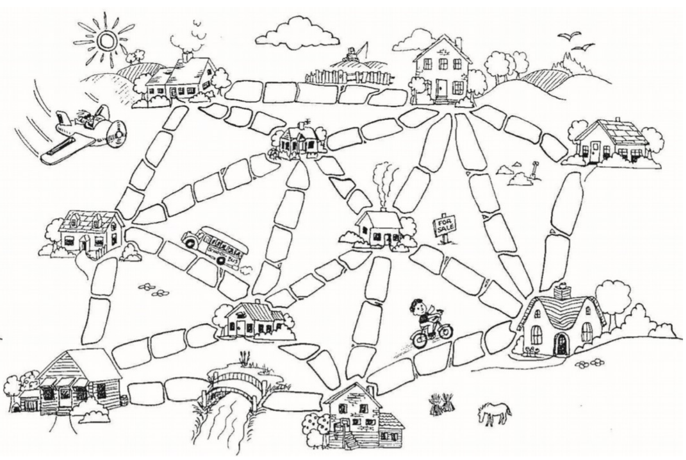
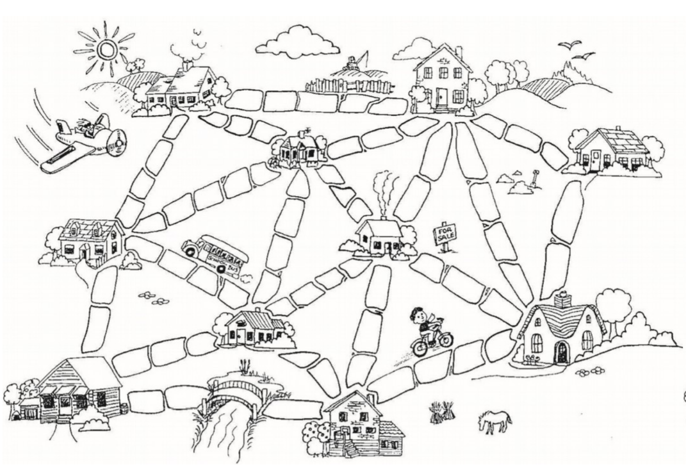
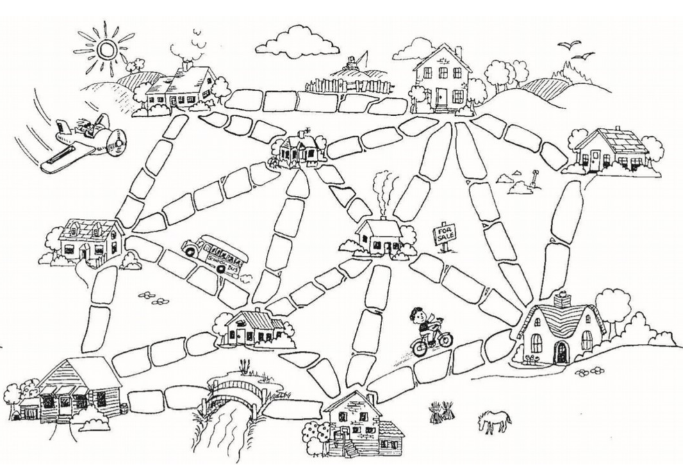
Le travail qui consiste à créer un réseau dont la longueur totale soit la plus courte possible est appelé problème de l'*arbre couvrant minimum.*

Les arbres couvrants minimaux ne sont pas seulement utiles pour les réseaux de gaz ou d'électricité, mais ils nous aident également à résoudre certains problèmes relatifs aux réseaux informatiques, téléphoniques, aux oléoducs ou aux voies aériennes. Cependant, lorsque vous choisissez l'itinéraire d'un voyage, vous devez prendre en compte l'aspect pratique du déplacement pour le voyageur tout autant que le coût. Personne ne veut passer des heures dans un avion qui fait un long détour par un autre pays sous prétexte que le billet est moins cher. L'algorithme de la ville sous la boue n'est pas d'un grand secours pour ces réseaux parce qu'il réduit simplement la longueur totale des routes ou des voies aériennes.

Les arbres couvrants sont également utiles dans la résolution d'autres problèmes sur les graphes tels que celui du « voyageur de commerce » qui essaie de trouver l'itinéraire le plus court pour passer par tous les points d'un réseau.

Il existe des algorithmes (méthodes) efficaces pour résoudre les problèmes des arbres couvrants minimaux. Une méthode simple donnant une solution optimale consiste à commencer sans aucune connexion puis à ajouter, dans un ordre de grandeur croissant, uniquement celles qui relient une partie de réseau qui n'était pas connectée auparavant. Il s'agit de l'algorithme de Kruskal, d'après le nom de J.B. Kruskal qui l'a publié en 1956.

Pour de nombreux autres problèmes de graphes, les informaticiens n'ont pas encore trouvé de méthodes suffisamment rapides dont les solutions soient les meilleures possibles pour résoudre les problèmes liés aux graphes, y compris celui du « voyageur de commerce ».


FOR SALE

FOR SALE

FOR SALE

FOR SALE

FOR SALE

FOR SALE

FOR SALE

FOR SALE