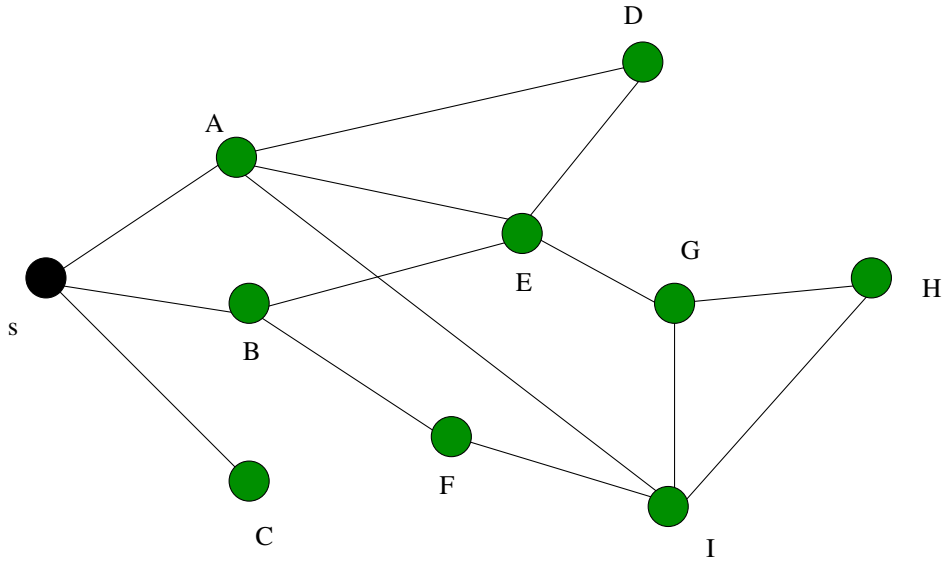
La recherche en largeur BFS (Breath First Search)

1. Dans l'exploration l'algorithme cherche à épuiser la liste des sommets proches de s avant de poursuivre l'exploration du graphe.

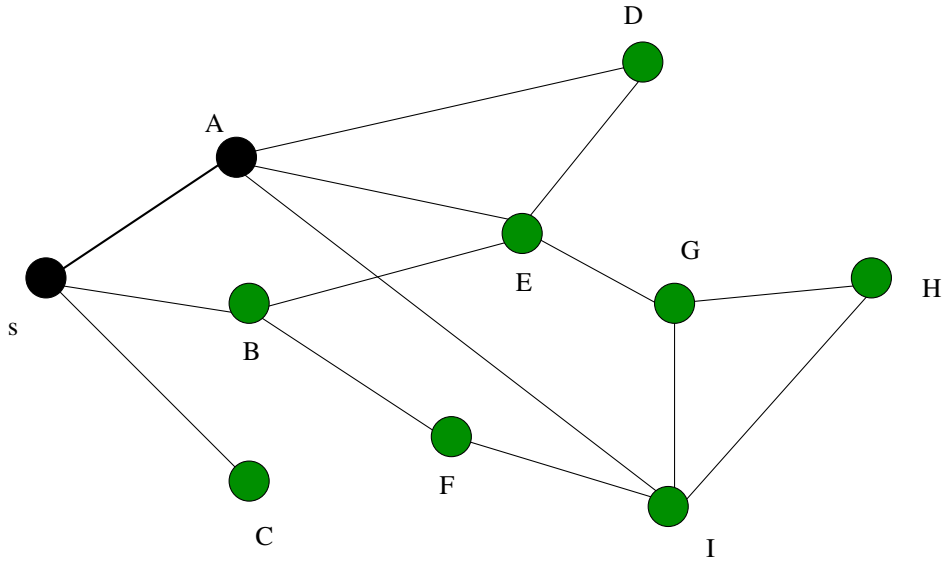
Exemple: BFS



La recherche en largeur BFS (Breath First Search)

1. Dans l'exploration l'algorithme cherche à épuiser la liste des sommets proches de s avant de poursuivre l'exploration du graphe.

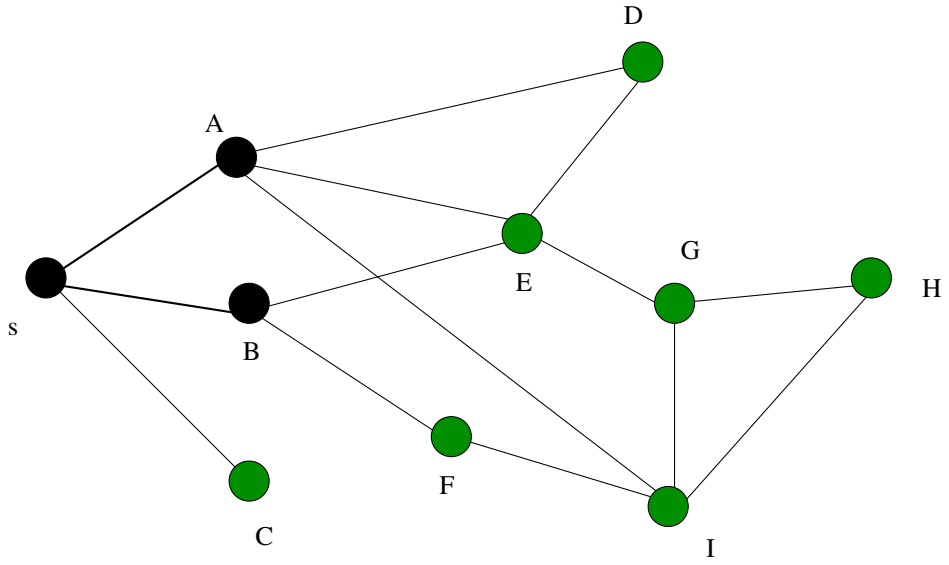
Exemple: BFS



La recherche en largeur BFS (Breath First Search)

1. Dans l'exploration l'algorithme cherche à épuiser la liste des sommets proches de s avant de poursuivre l'exploration du graphe.

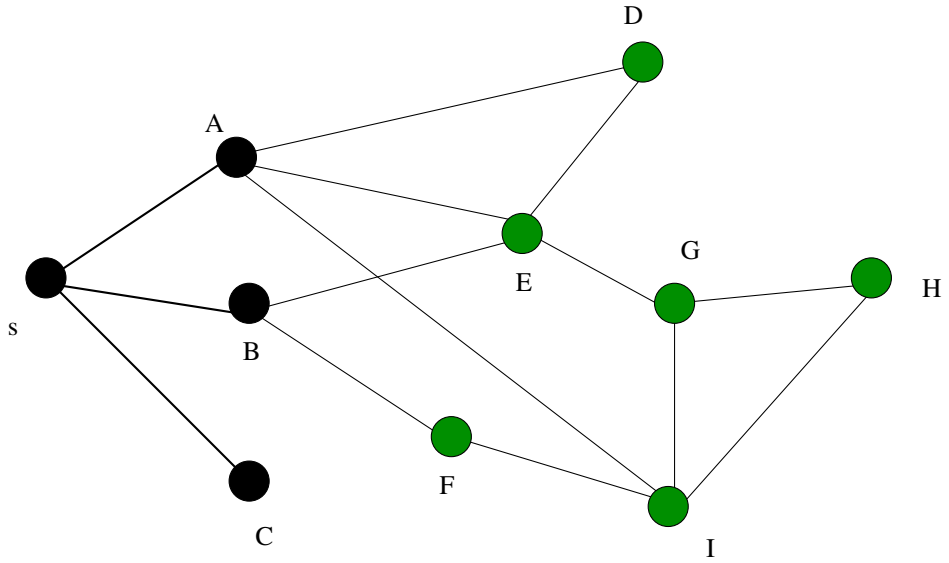
Exemple: BFS



La recherche en largeur BFS (Breath First Search)

1. Dans l'exploration l'algorithme cherche à épuiser la liste des sommets proches de s avant de poursuivre l'exploration du graphe.

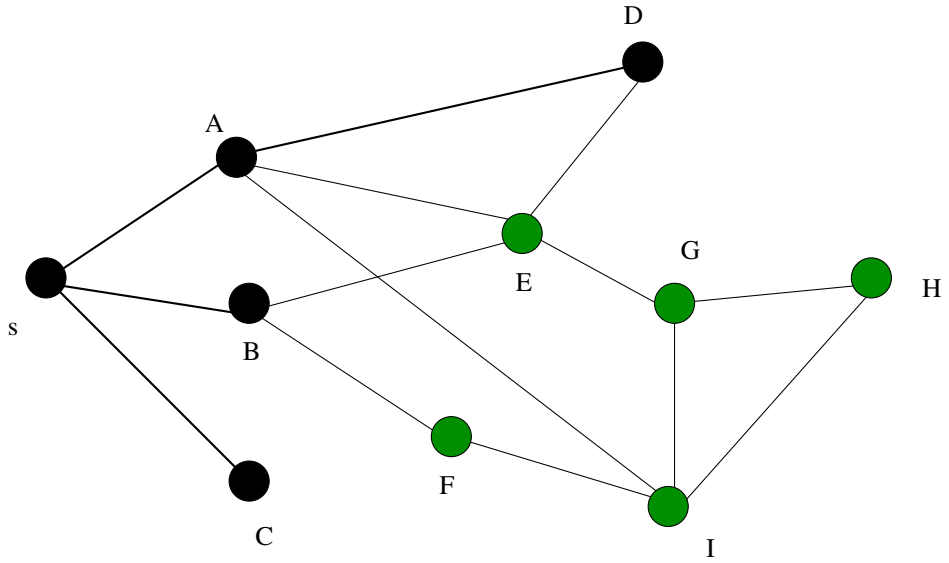
Exemple: BFS



La recherche en largeur BFS (Breath First Search)

1. Dans l'exploration l'algorithme cherche à épuiser la liste des sommets proches de s avant de poursuivre l'exploration du graphe.

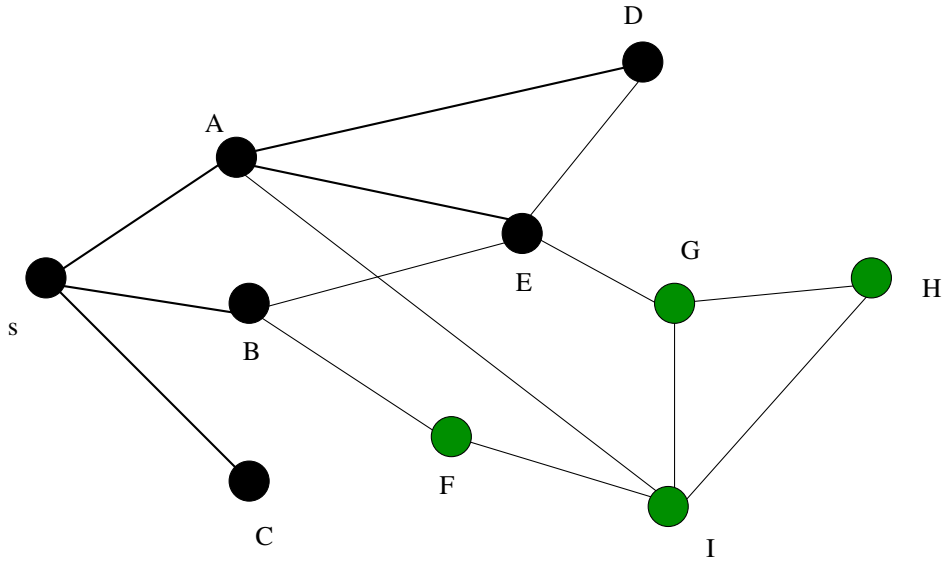
Exemple: BFS



La recherche en largeur BFS (Breath First Search)

1. Dans l'exploration l'algorithme cherche à épuiser la liste des sommets proches de s avant de poursuivre l'exploration du graphe.

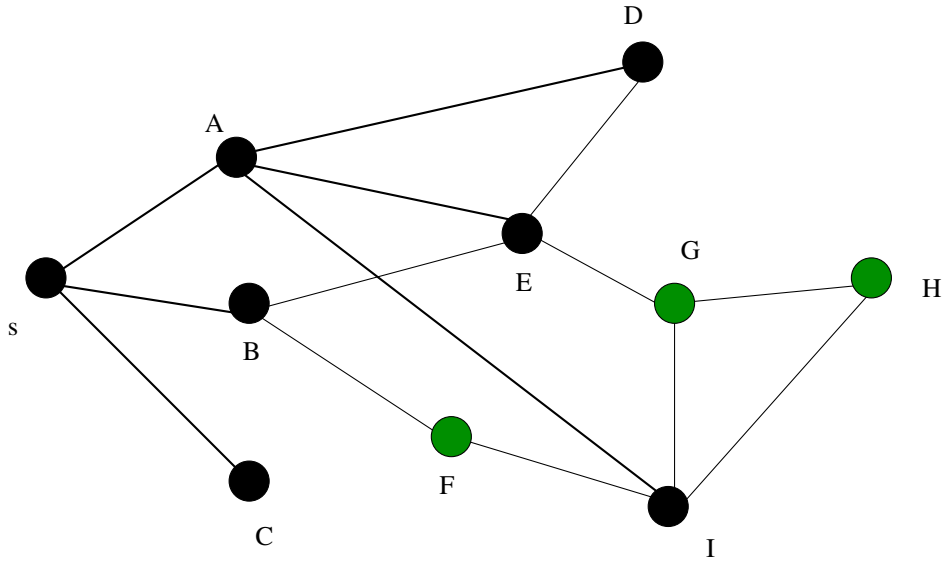
Exemple: BFS



La recherche en largeur BFS (Breath First Search)

1. Dans l'exploration l'algorithme cherche à épuiser la liste des sommets proches de s avant de poursuivre l'exploration du graphe.

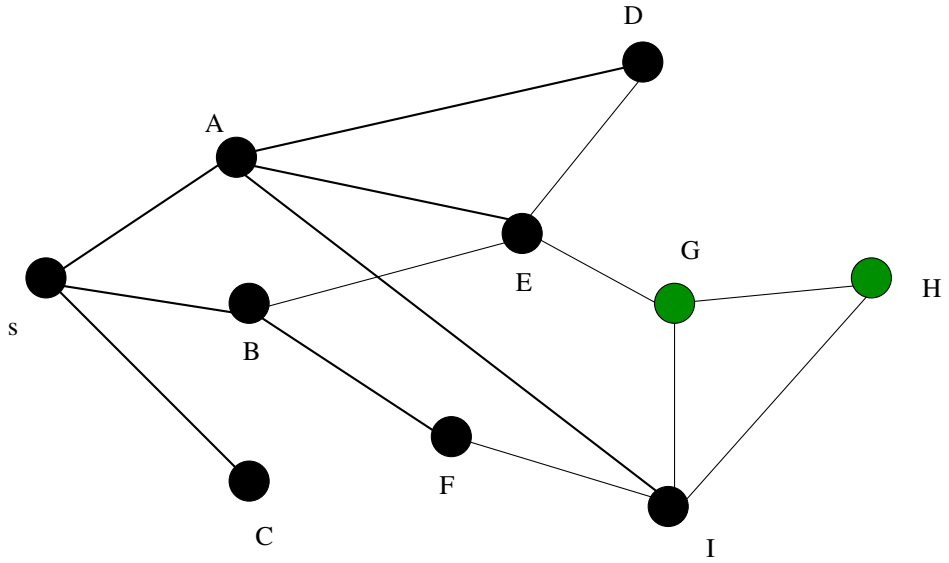
Exemple: BFS



La recherche en largeur BFS (Breath First Search)

1. Dans l'exploration l'algorithme cherche à épuiser la liste des sommets proches de s avant de poursuivre l'exploration du graphe.

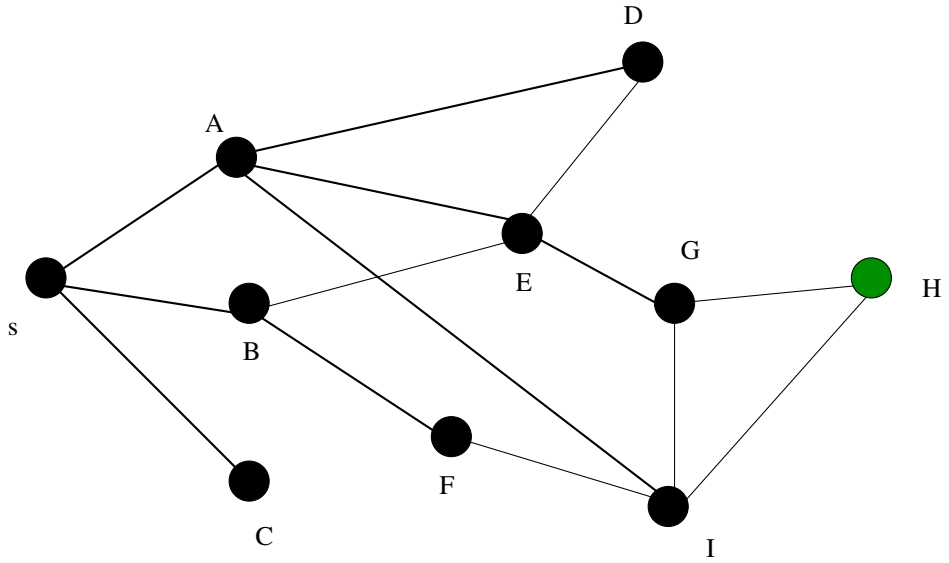
Exemple: BFS



La recherche en largeur BFS (Breath First Search)

1. Dans l'exploration l'algorithme cherche à épuiser la liste des sommets proches de s avant de poursuivre l'exploration du graphe.

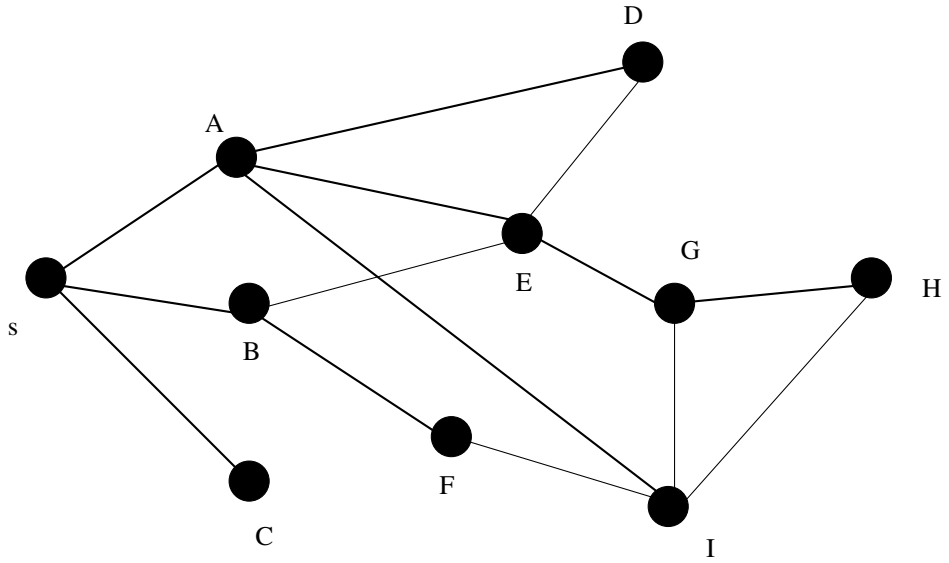
Exemple: BFS



La recherche en largeur BFS (Breath First Search)

1. Dans l'exploration l'algorithme cherche à épuiser la liste des sommets proches de s avant de poursuivre l'exploration du graphe.

Exemple: BFS



Analyse du BFS

Le BFS correspond bien à l'ordre de visite du graphe voulu pour résoudre le problème de plus courts chemins.

Synthèse du BFS

1. Soit M une structure de données contenant l'ensemble des sommets marqués ayant au moins un voisin non traité.
2. A chaque étape, sélectionner un sommet x de M (en le retirant de M), marquer tous ses voisins (non encore marqués) et ajouter les à M.

Analyse: La structure de données M posséde 2 actions :

- l'ajout d'un sommet x à M, et
- le retrait du sommet en tête de la structure M.

Implémentation

La Recherche en Profondeur correspond à prendre pour M une structure de *PILE*, c'est à dire une liste *LIFO* LastIn/FirstOut (dernier entré/premier sorti)
La Recherche en Largeur correspond à prendre pour M une structure de *FILE*, c'est à dire une liste *FIFO* FirstIn/FirstOut (premier entré/premier sorti)

ALGORITHME BFS(G: Graphe, S : Sommet)

F : File de sommets;

Initialiser(G);

S.couleur = Noir;

$L(S) = 0$;

Ajouter S à F;

tant que F n'est pas vide faire

    x = Defiler(F);
    pour tous les successeurs y de x faire si y.couleur == Blanc alors

        y.couleur ¡- Noir;
        $L(y) = L(x) + 1$
        Ajouter y à F;

    fin pour

fin tant que

Theorème

L'algorithme *BFS* calcule en temps $O(|E|)$ la longueur des plus courts chemins du sommet s à tous les autres sommets du graphe, c-a-d pour tout x, $L(x) = D(x)$.

Theorème

L'algorithme *BFS* calcule en temps $O(|E|)$ la longueur des plus courts chemins du sommet s à tous les autres sommets du graphe, c-a-d pour tout x, $L(x) = D(x)$.

Preuve.

Il est clairement que $L(x)$ correspond à la longueur du chemin dans l'arbre de recherche construit par BFS. Nous avons donc $L(x) \geqslant D(x)$.

Theorème

L'algorithme *BFS* calcule en temps $O(|E|)$ la longueur des plus courts chemins du sommet s à tous les autres sommets du graphe, c-a-d pour tout x, $L(x) = D(x)$.

Preuve.

Il est clairement que $L(x)$ correspond à la longueur du chemin dans l'arbre de recherche construit par BFS. Nous avons donc $L(x) \geqslant D(x)$. On peut également montrer par induction que si un sommet u est ajouté à la file F avant un sommet v, alors on a $L(u) \leqslant L(v)$.