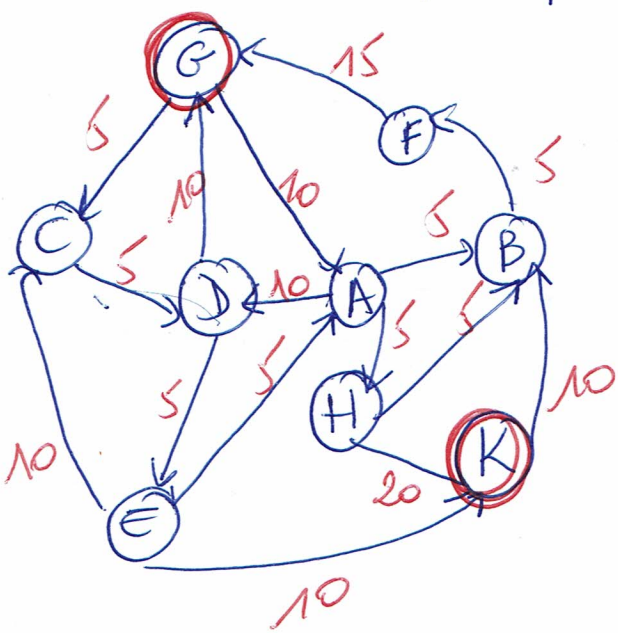
Méthode de recherche d'une solution dans un espace d'état

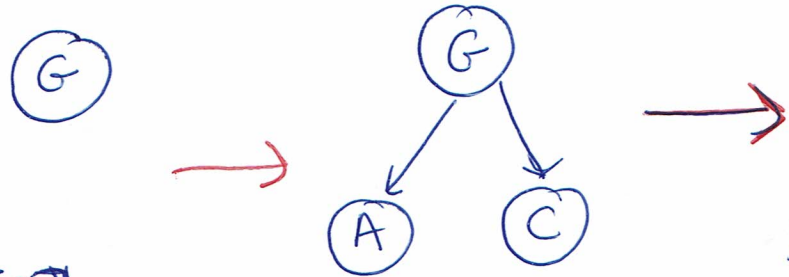
Recherche en profondeur d'abord (Depth first)

Recherche aveugle

Nous allons reprendre le même problème que celui de la recherche en largeur d'abord.
La méthode Depth first est plus efficace, car un état visité ne peut pas être visité une autre fois.



On veut aller de la ville G à la ville K en utilisant la méthode en profondeur d'abord.

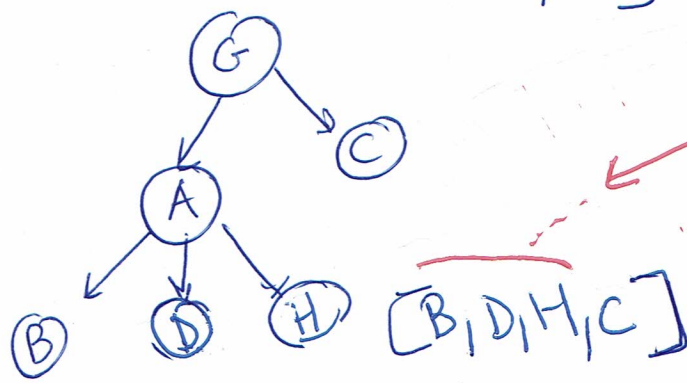


On explore chaque niveau en terminant avec sa profondeur. ici le A sera remplacé par son expansion.

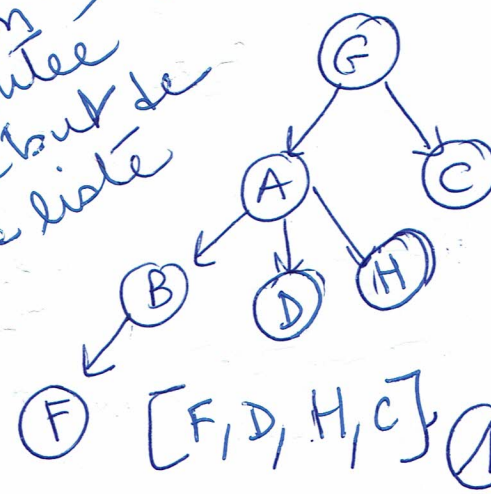
[G]

[A, C]

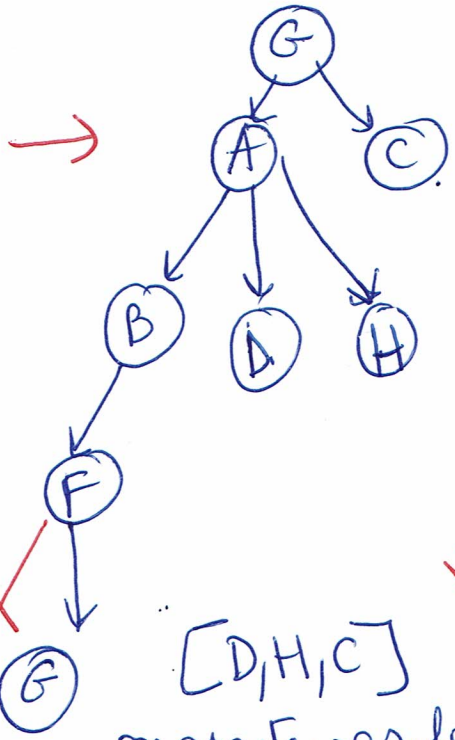
L'expansion sera ajoutée au début de la liste



[B, D, H, C]



[F, D, H, C]

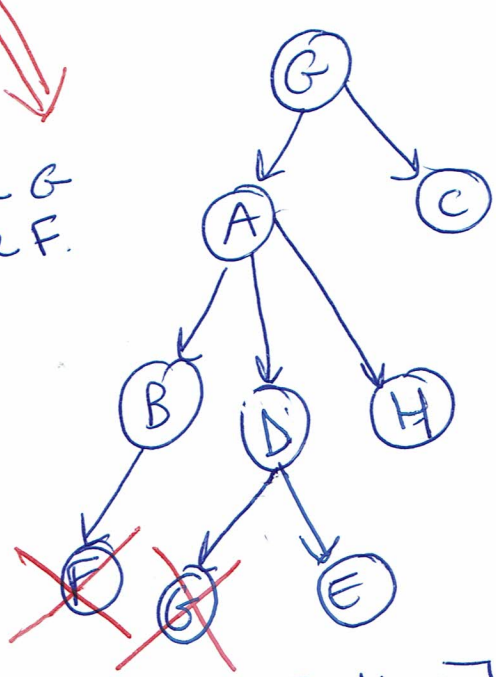


[D, H, C]

on ajoute pas le G et on retire le F.

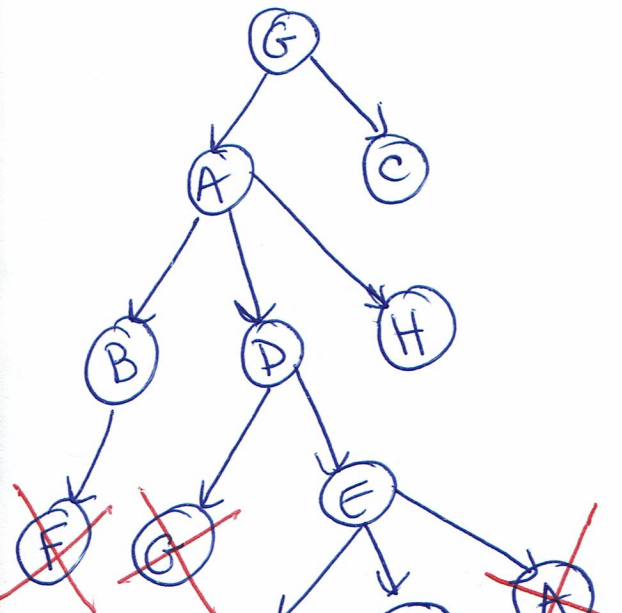
On va pas le retenir car F amène à G et G a été déjà visité on passe donc à H.

la recherche en profondeur est une solution pour remédier aux problèmes de la recherche en largeur.
les noeuds visités déjà ne seront pas pris en considération.



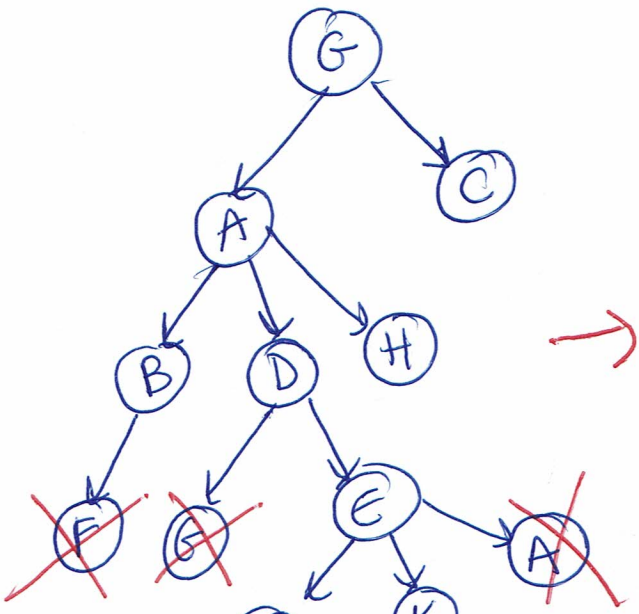
[E, H, C]

On continue en profondeur toujours avec E maintenant

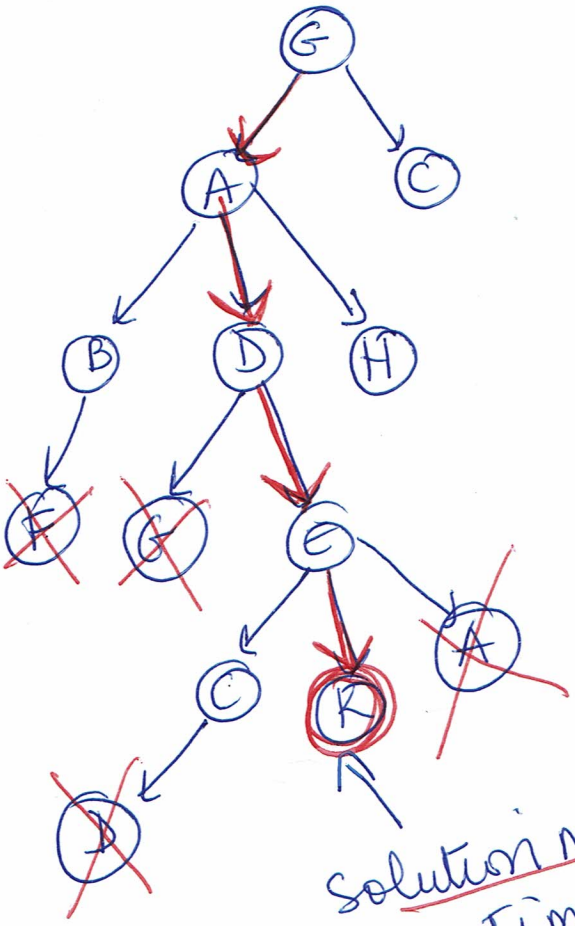


déjà visité

[C, K, H, C]



[K, H, C]



le chemin G → A → D → E → K

Cette solution est plus optimale que la recherche en largeur d'abord.

cet algorithme peut économiser la mémoire.

[H/C]

l'espace mémoire utilisé est proportionnel à la profondeur dans l'arbre.

Cette solution ne peut garantir une solution optimale surtout si les coûts sont différents.

⟹ nous allons voir donc d'autres algorithmes plus intéressants (algorithmes heuristiques)