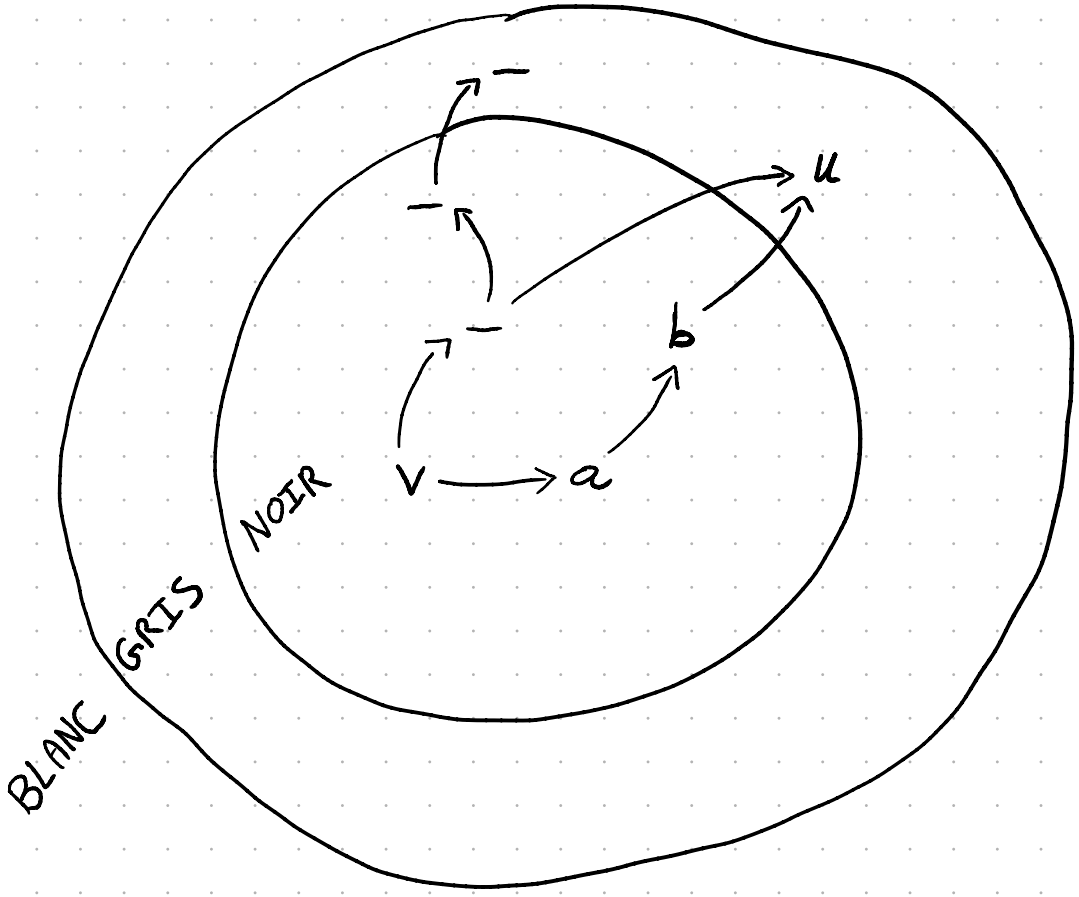


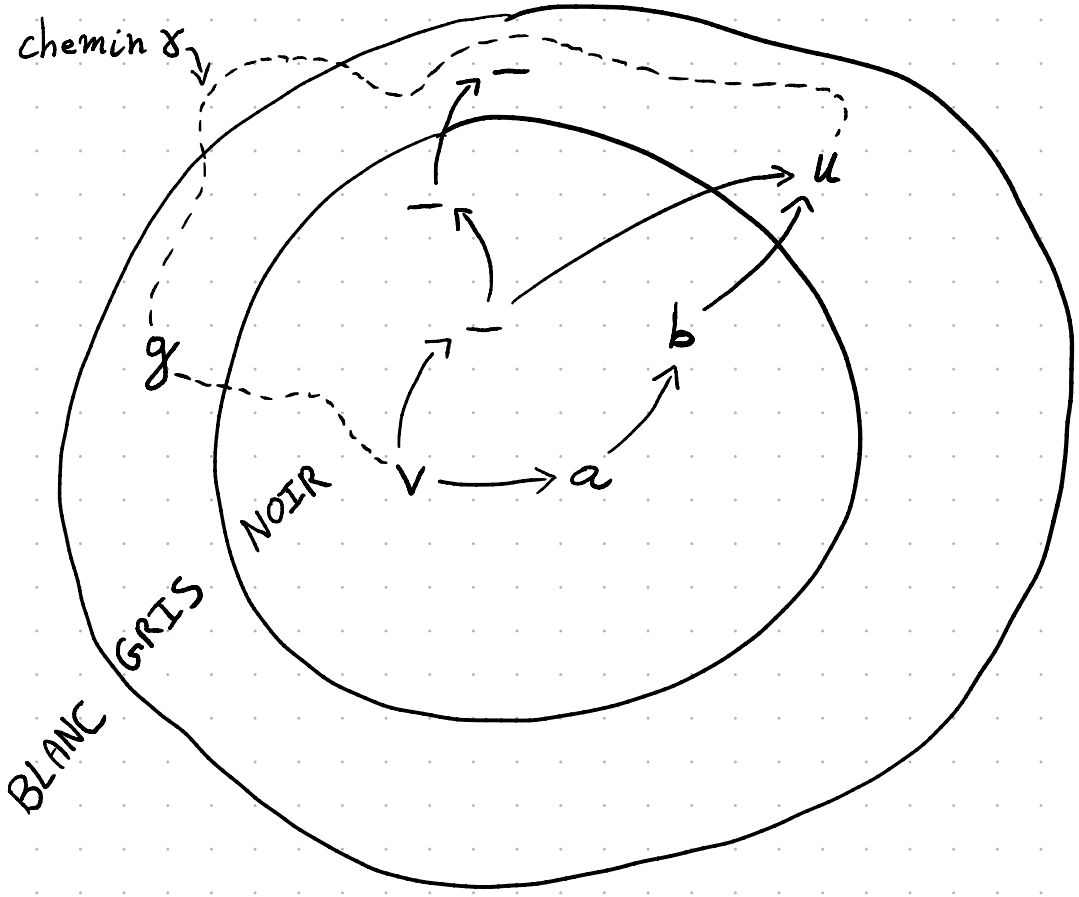
$\forall x \in$ NOIR, $d[x]$ = longueur du plus court chemin reliant la source v à x.

Le plus court chemin de v à x est fait de nœuds noirs.

$\forall x \in$ GRIS, $d[x]$ = longueur du plus court chemin jusqu'ici découvert pour aller de la source v à x.

Ce chemin est fait de nœuds noirs, sauf x qui est gris.

Tout chemin de v à un nœud blanc passe par un nœud gris.



Soit u un nœud gris qui minimise $d[u]$.
Par exemple, $d[u] = w(v,a) + w(a,b) + w(b,u)$
avec $w(x,y)$ le poids de l'arête $x \to y$.

Soit un chemin γ quelconque de v à u, de longueur ℓ.

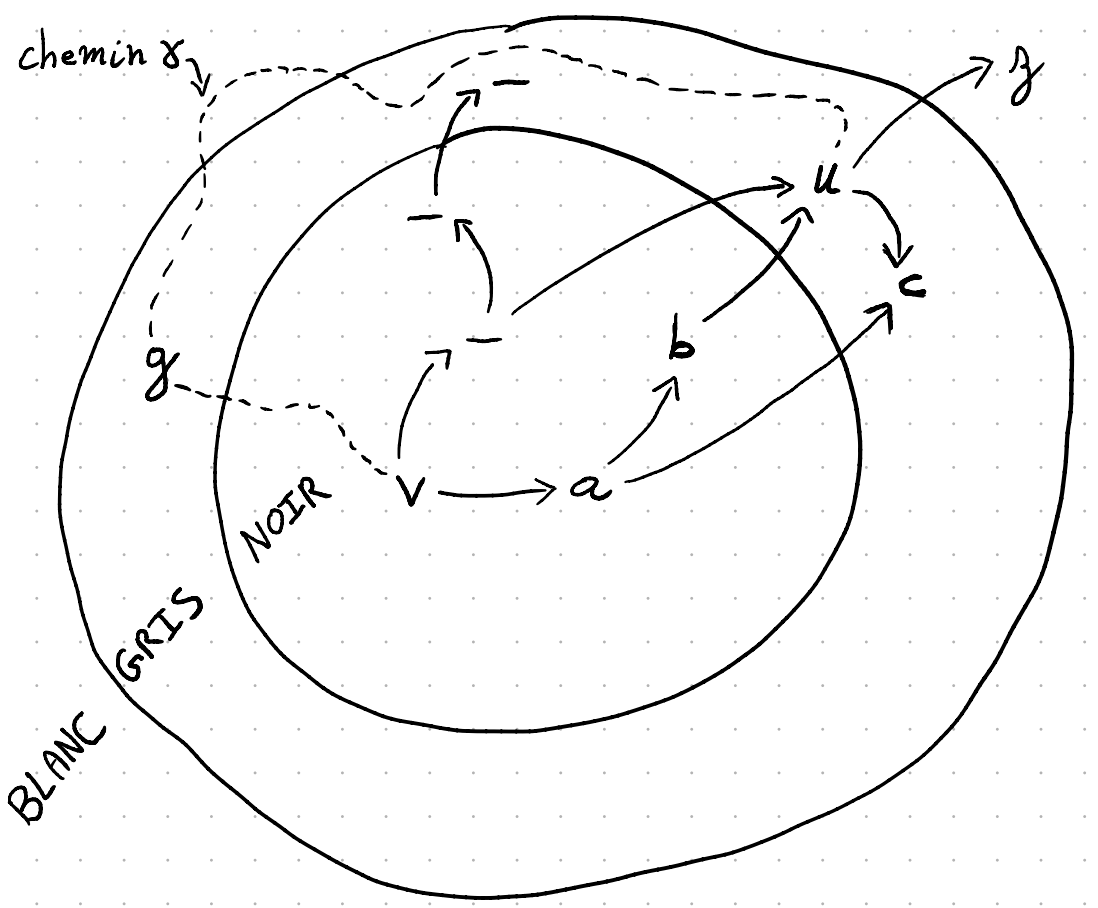
Soit g le premier nœud gris sur ce chemin.

$$d[u] \leqslant d[g]$$

La distance le long de γ séparant g de u est positive.

$$d[g] \leqslant \ell$$

Donc $d[u] \leqslant \ell$ et $d[u]$ est la longueur du plus court chemin reliant v à u.

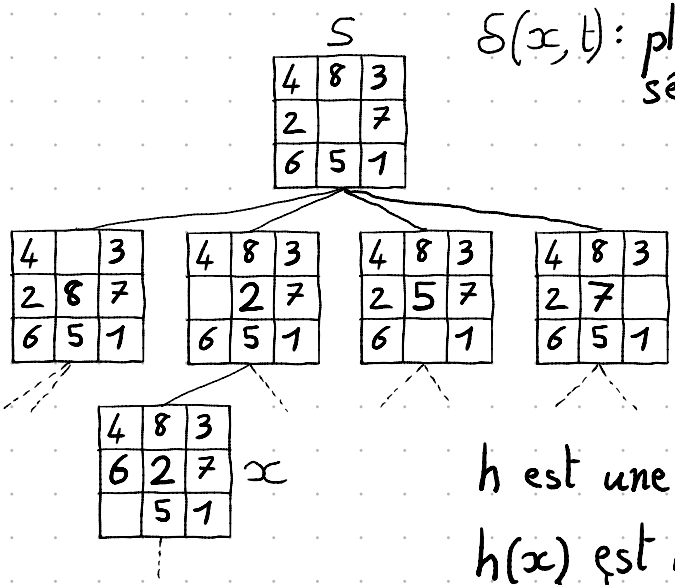


Pour que u devienne noir, il faut mettre à jour les distances $d[\,]$ associées à ses successeurs.

Soit z un successeur blanc de u,
il devient gris avec $d[z] = d[u] + w(u,z)$

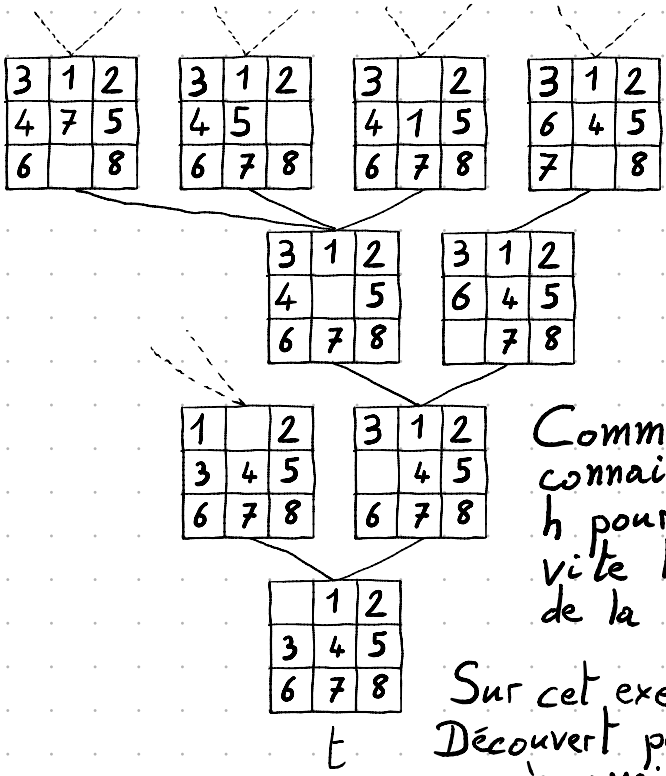
Soit c un successeur gris de u,
$d[c]$ est mis à jour si $d[c] \geqslant d[u] + w(u,c)$

$\delta(x,t)$: plus courte distance séparant x de t.



h est une fonction heuristique

$h(x)$ est une estimation de la distance séparant x de t.



Comment utiliser la connaissance apportée par h pour découvrir plus vite le plus court chemin de la source vers la cible ?

Sur cet exemple, $\delta(s,t) = 20$.
Découvert par recherche en largeur après avoir explorées 44 696 configurations

Peut-on employer l'algorithme de recherche de plus court chemin de Dijkstra en associant à un nœud u une sous-estimation de la longueur du chemin qui, partant de la source, passe par u et rejoint la cible ?

$$F(u) = g(u) + h(u)$$

longueur du plus court chemin jusqu'à présent découvert pour rejoindre u depuis le nœud de départ s.

Soit u le plus petit nœud gris au sens de F.
Soit v un successeur de u.
A-t-on découvert une meilleure sous-estimation de la longueur du plus court chemin qui passe par v et rejoint la cible ?

$$
\begin{aligned}
& F(v) \\
= & \{\text{par définition de } F\} \\
& g(v) + h(v) \\
= & \{\text{si on rejoint } v \text{ depuis } u\} \\
& g(u) + w(u,v) + h(v) \\
= & \\
& g(u) + h(u) + w(u,v) + h(v) - h(u) \\
= & \{\text{définition de } F\} \\
& F(u) + w(u,v) + h(v) - h(u)
\end{aligned}
$$

Si la valeur $f(v)$ actuellement associée à v est plus grande que : $f(u) + w(u,v) + h(v) - h(u)$, alors, elle est mise à jour.

Pour que l'algorithme de Dijkstra soit applicable, il faut que les poids des arêtes soient positifs.

$$h(v) + w(u,v) \geqslant h(u)$$

On dit que h est cohérente.

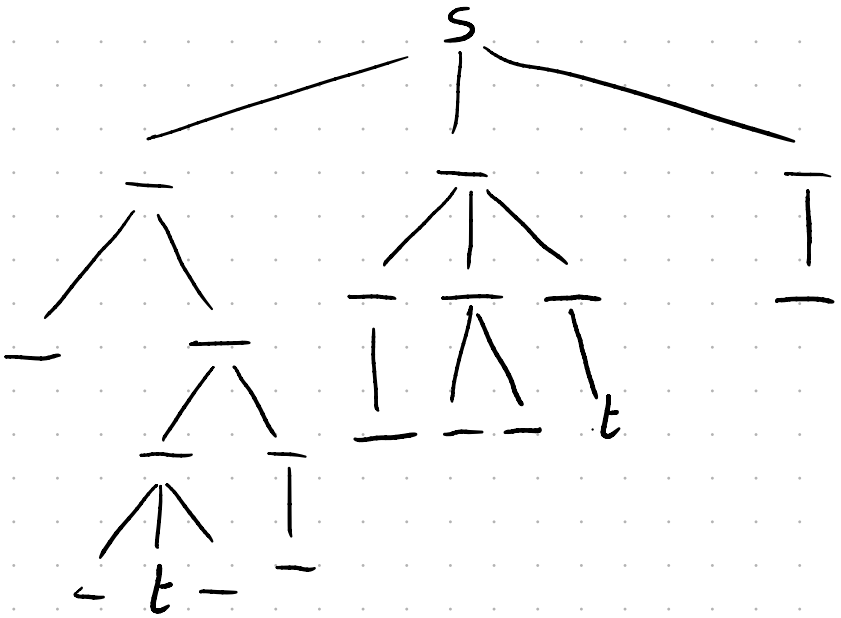
Ce qui est plus fort que seulement dire que h est admissible (i.e., sous-estime toujours la distance à la cible).

4	8	3
2		7
6	5	1

Solution optimale en 20 coups.

	nb configurations explorées :
$h = 0$	44696
$h =$ "nb pièces mal placées"	2877
$h =$ "somme distances Manhattan"	189

L'optimalité des algorithmes de recherche en largeur dépend de la mémorisation de toutes les configurations de coûts inférieurs à la solution.

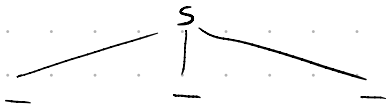


Comment utiliser la recherche en profondeur, qui ne mémorise que le chemin courant, tout en assurant la découverte du plus court chemin vers la cible ?

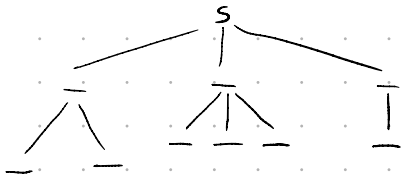
Approfondissement itéré.

Profondeur maximum

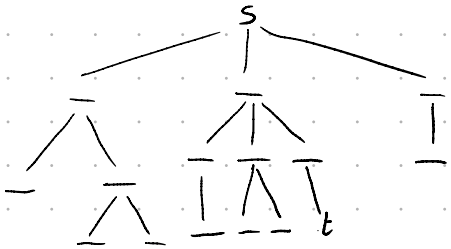
1



2

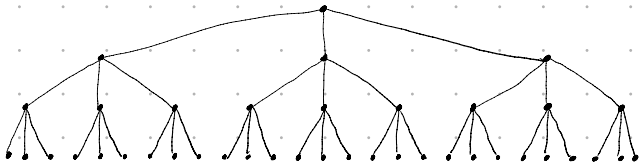


3



Facteur de branchement : nb moyen de successeurs d'une configuration.

Exemple d'arbre de recherche pour un facteur de branchement de 3 :



1
3
9
27

Quel est le nombre de configurations explorées à chaque itération d'un approfondissement itéré ?

$1+3=4$ configurations explorées à l'itération 1

$4+9=13$ ——— 2

$13+27=40$ ——— 3

Le nb de configurations explorées durant la dernière itération (ici 40) domine largement la somme des configurations explorées durant les itérations précédant la dernière (ici $13+4=17$).

IDA : approfondissement itéré appliqué à la stratégie A* (i.e. Dijkstra + heuristique).

À chaque itération, la profondeur maximum est une valeur de $F = g + h$.

À la première itération, la profondeur max est

$$F(s) = g(s) + h(s) = h(s)$$

Lors de la première itération, quand une configuration rencontrée x est telle que $F(x) > F(s)$, ses successeurs ne sont pas explorés.

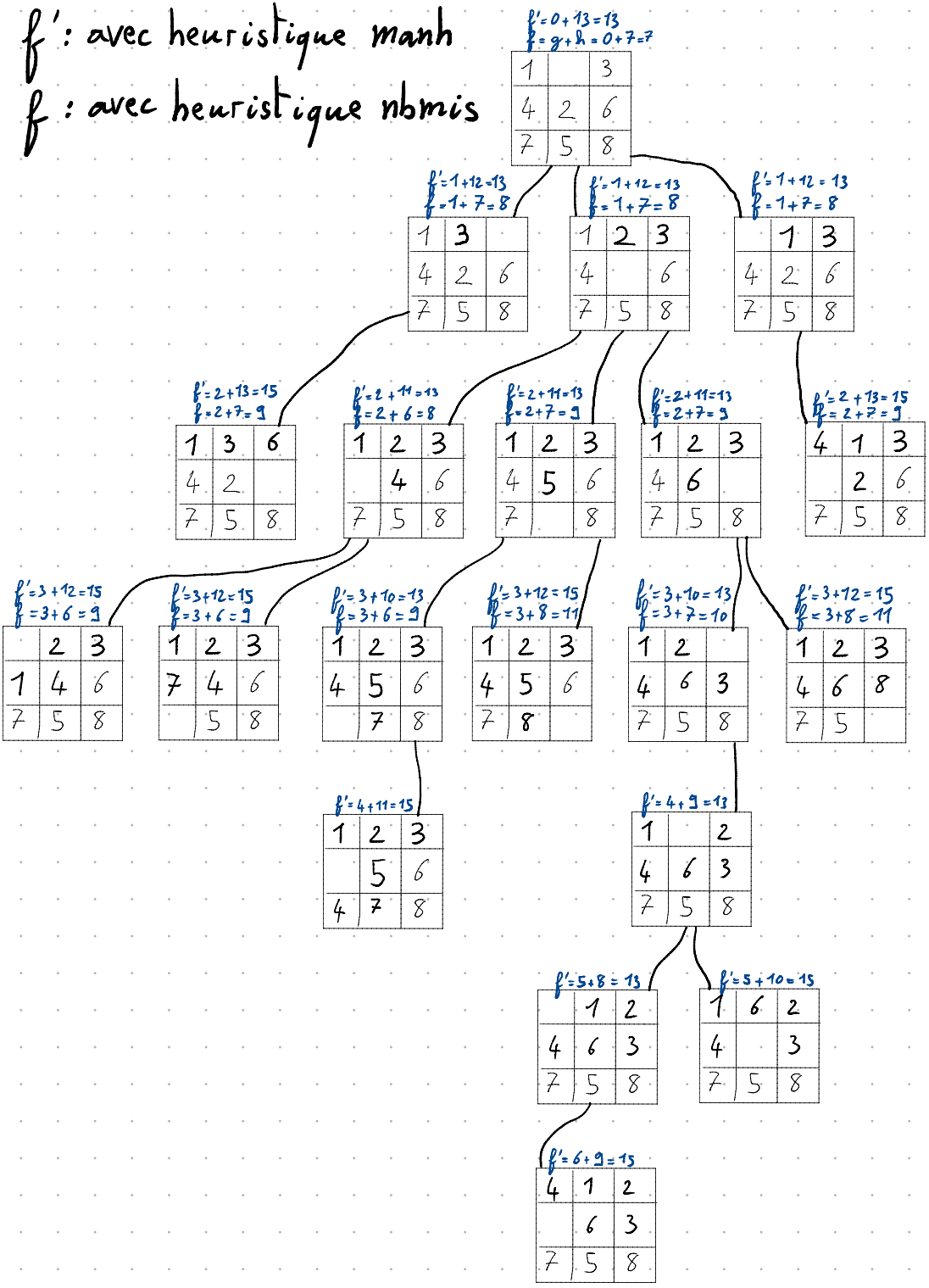
On mémorise la valeur $F(x^*)$ de la configuration x^* qui a le moins dépassé la profondeur max de l'itération courante.

Cette valeur devient la profondeur max de la prochaine itération.

Etc.

f' : avec heuristique manh

f : avec heuristique nbmis



ε-optimalité et weighted-IDA

$$
\begin{aligned}
& F'(u) \\
= \; & g(u) + (1+\varepsilon)\, h(u) \\
= \; & \{ u \text{ est un nœud gris minimal} \} \\
& \delta(s,u) + (1+\varepsilon)\, h(u) \\
\leqslant \; & \{ h \text{ est admissible} \} \\
& \delta(s,u) + \delta(u,t) + \varepsilon\, \delta(u,t) \\
= \; & \{ \text{avec } \gamma \text{ un plus court chemin de } s \text{ à } t \text{ passant par } u \} \\
& \gamma(s,t) + \varepsilon\, \delta(u,t) \\
\leqslant \; & (1+\varepsilon)\, \gamma(s,t)
\end{aligned}
$$

En particulier, $F'(t) \leqslant (1+\varepsilon)\, \delta(s,t)$.

Exemple 15-puzzle :

$\varepsilon = 1{,}5$: solution en 48 coups découverte après la visite de 23 326 configurations.

$\varepsilon = 1$: solution optimale en 46 coups découverte après la visite de 1 082 032 configurations