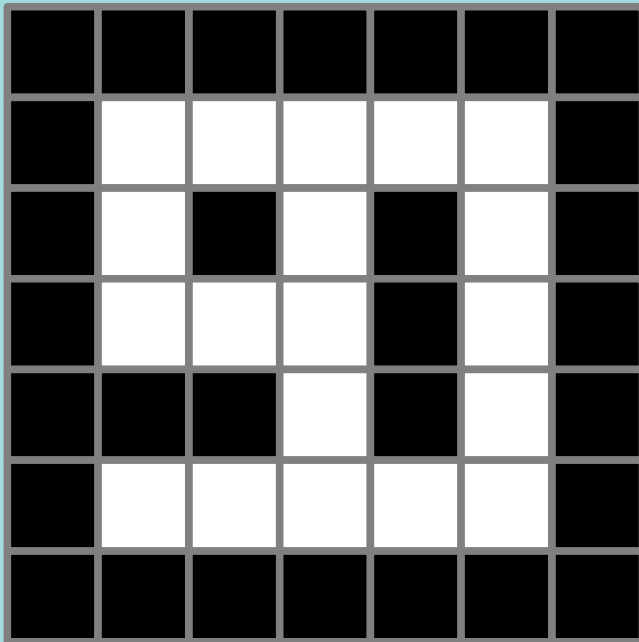
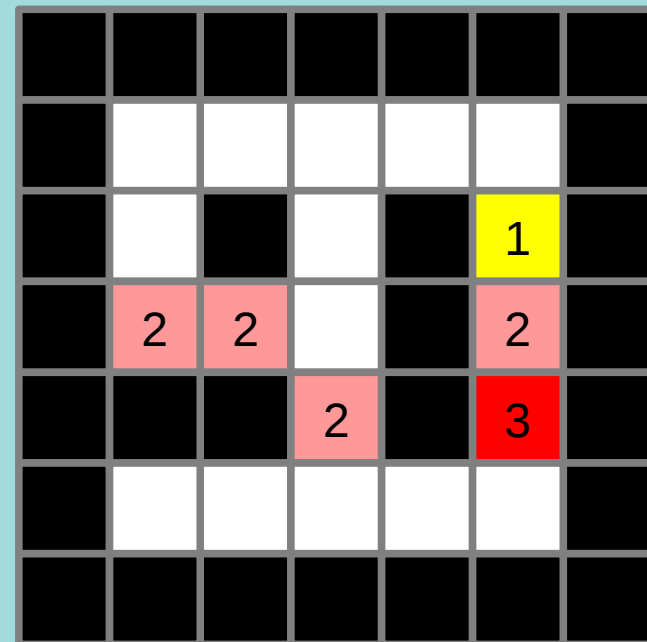


# Dijkstra

- Labyrinthe



Murs



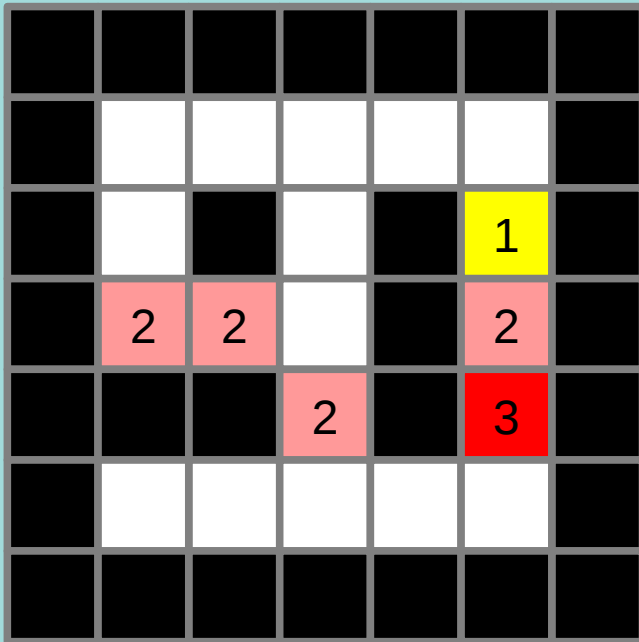
Cout = difference altitude

# Dijkstra

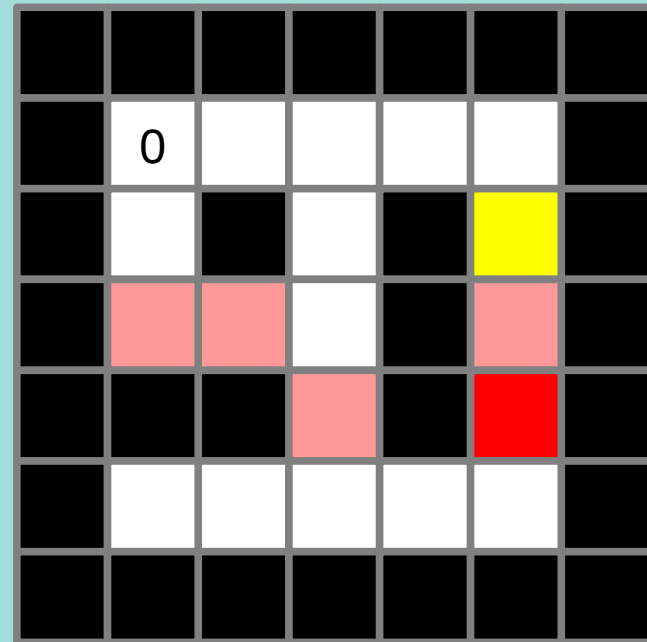
- Calcul des distances et Propagation

# Dijkstra

- Calcul des distances et Propagation



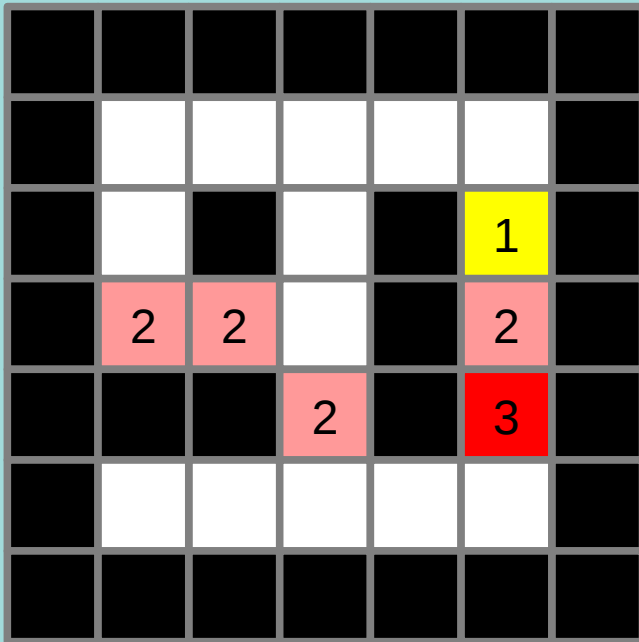
Cost = difference altitude



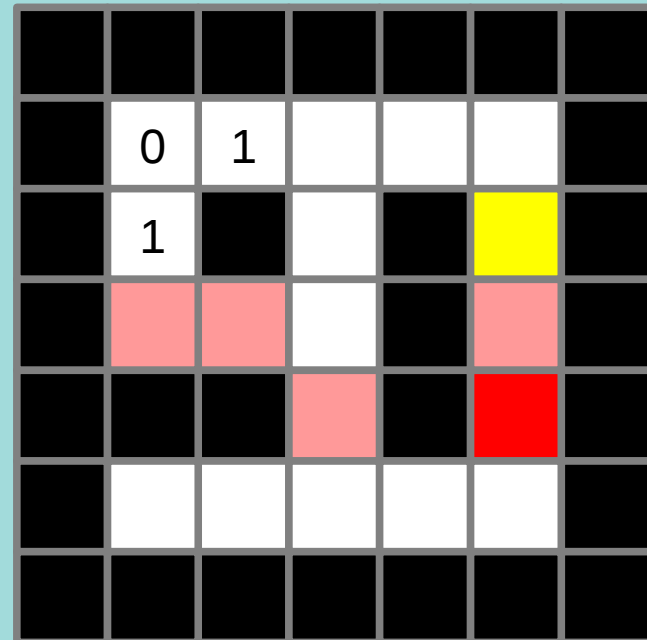
Distances

# Dijkstra

- Calcul des distances et Propagation



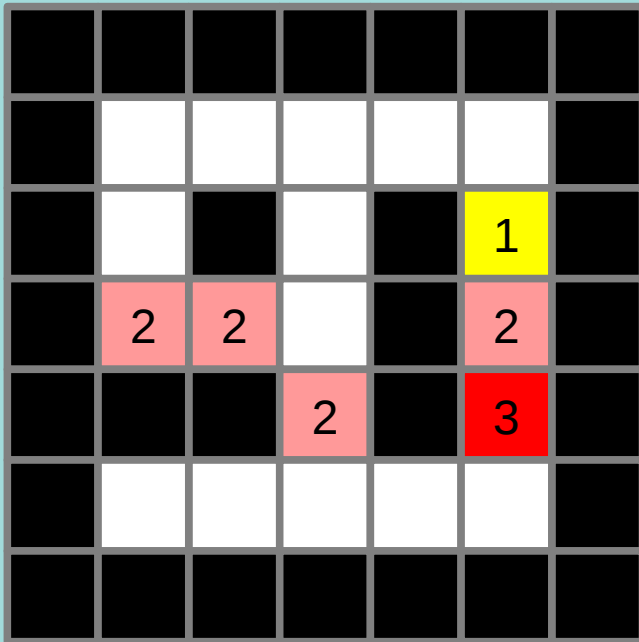
Cost = difference altitude



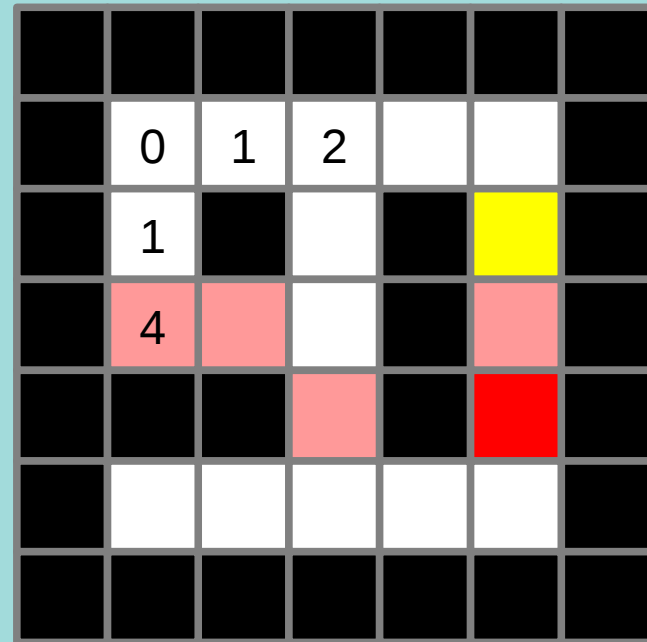
Distances

# Dijkstra

- Calcul des distances et Propagation



Cost = difference altitude

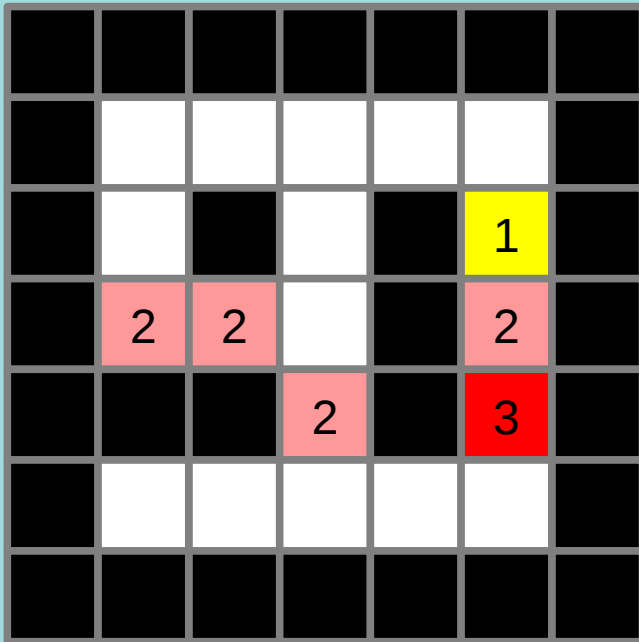


Distances

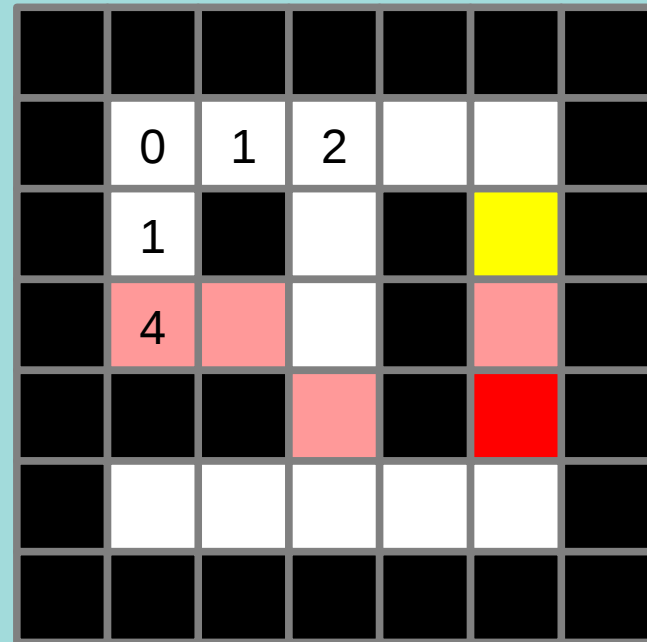
Comment développer ?

# Dijkstra

- Calcul des distances et Propagation



Cout = difference altitude

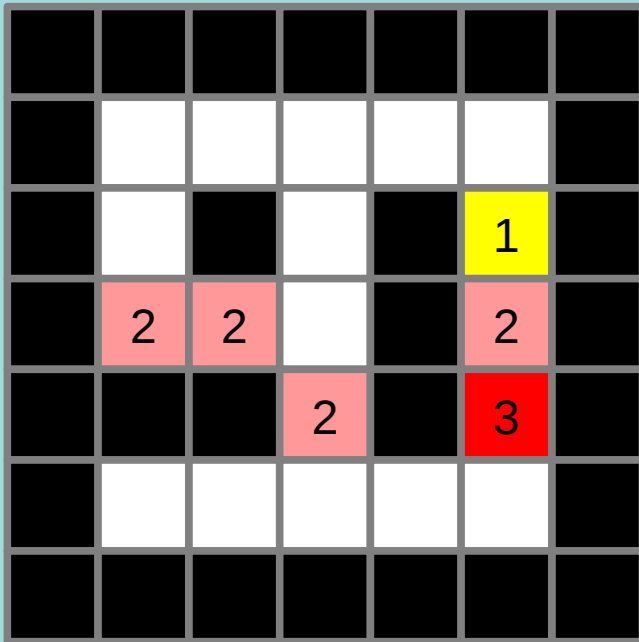


Distances

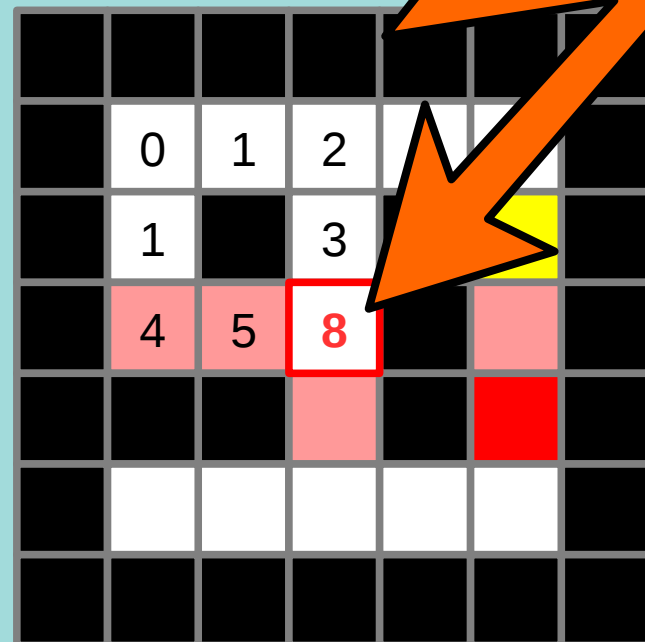
Dijkstra => plus proche d'abord

# Dijkstra

- Calcul des distances et Propagation



Coût = différence altitude

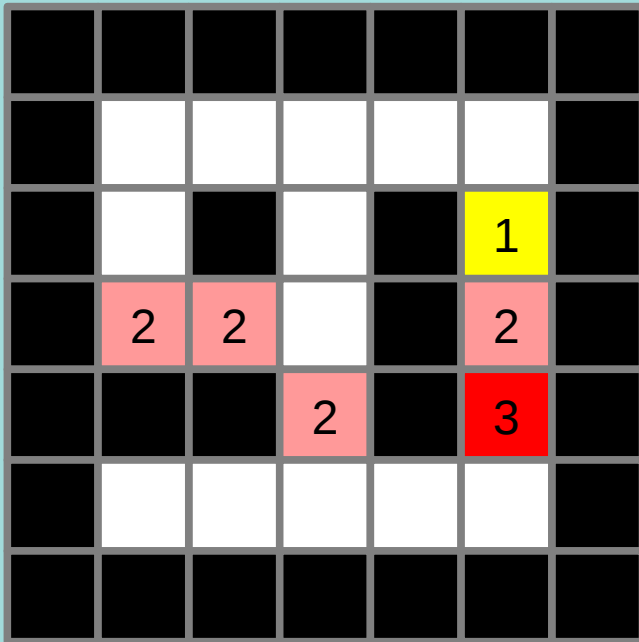


Distances

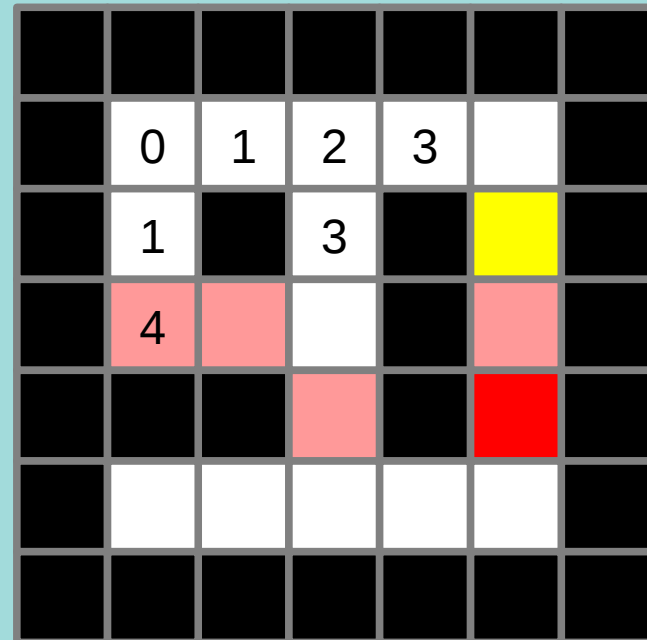
SINON....

# Dijkstra

- Calcul des distances et Propagation



Cout = difference altitude



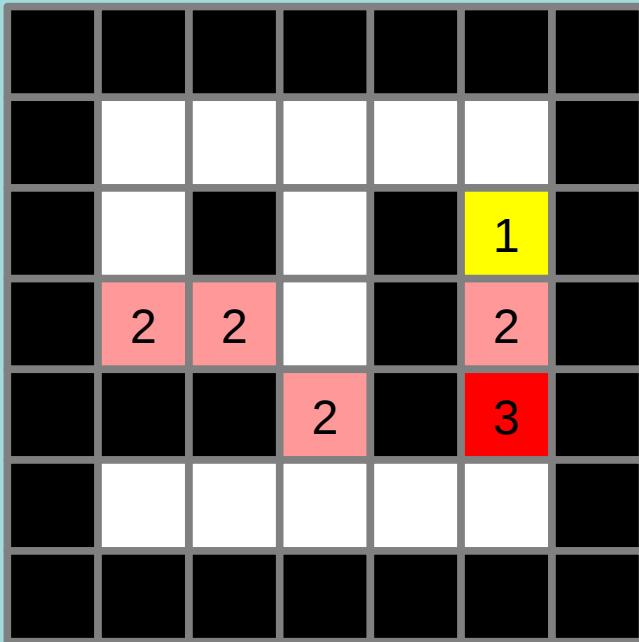
Distances

Dijkstra => plus proche d'abord

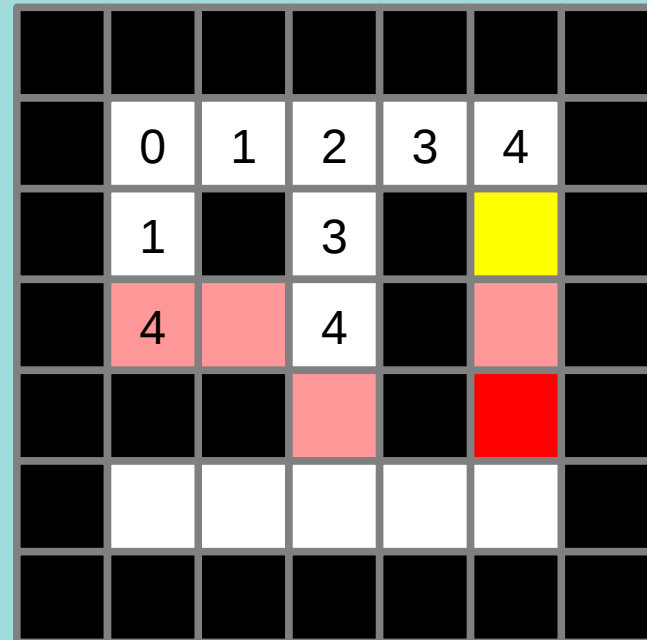


# Dijkstra

- Calcul des distances et Propagation



Cout = difference altitude

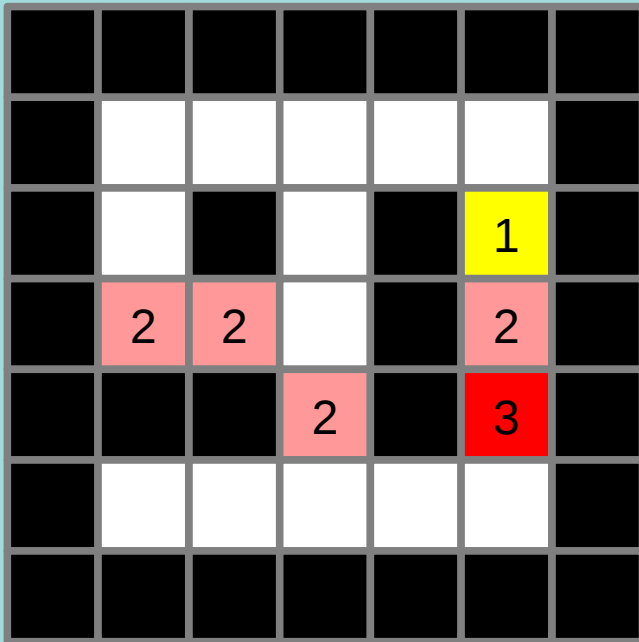


Distances

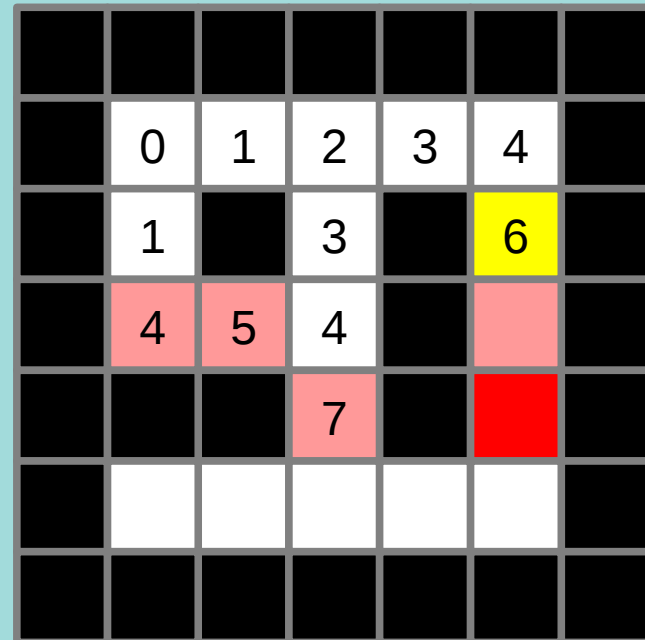
Dijkstra => plus proche d'abord

# Dijkstra

- Calcul des distances et Propagation



Coût = différence altitude

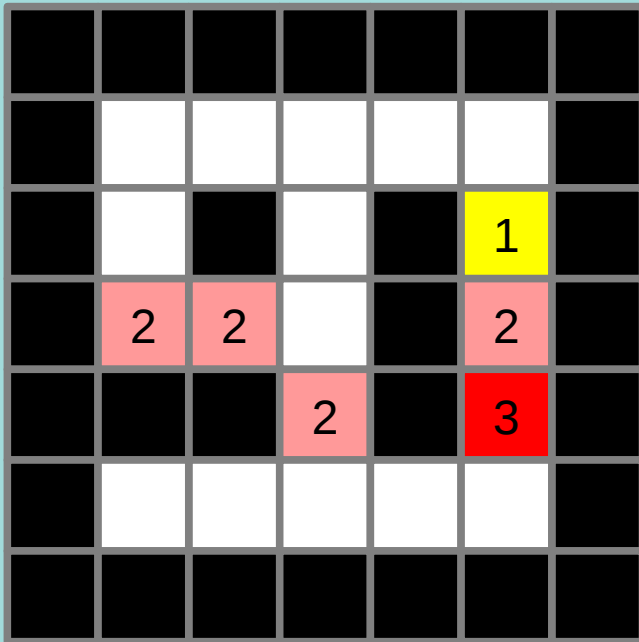


Distances

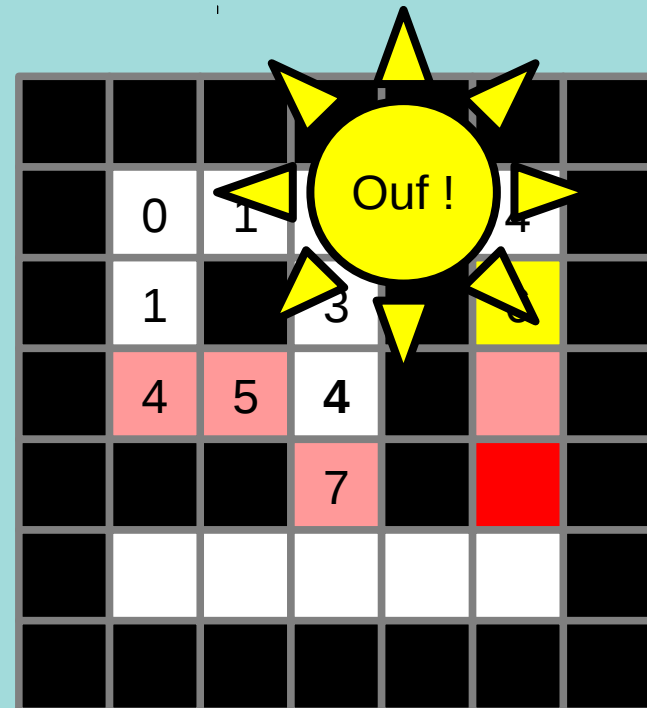
Dijkstra => plus proche d'abord

# Dijkstra

- Calcul des distances et Propagation



Cout = difference altitude

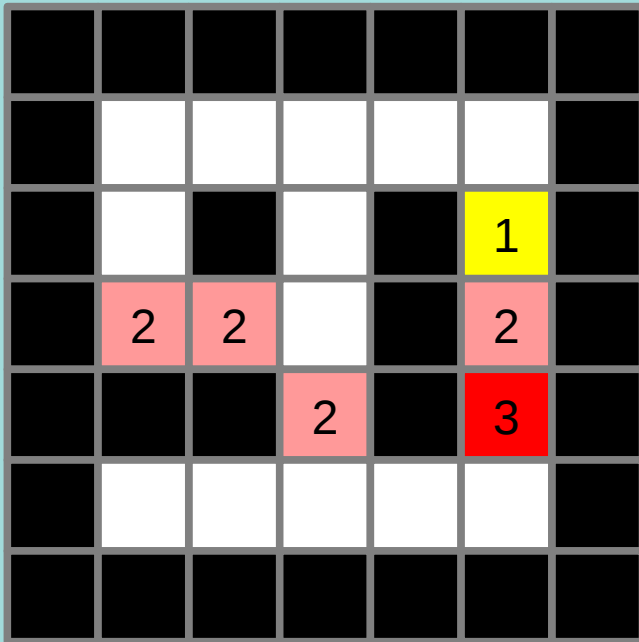


Distances

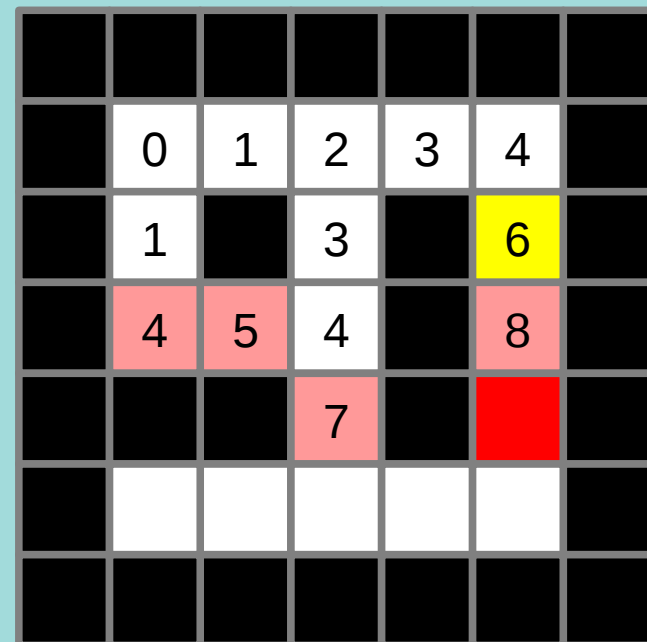
Dijkstra => plus proche d'abord

# Dijkstra

- Calcul des distances et Propagation



Coût = différence altitude

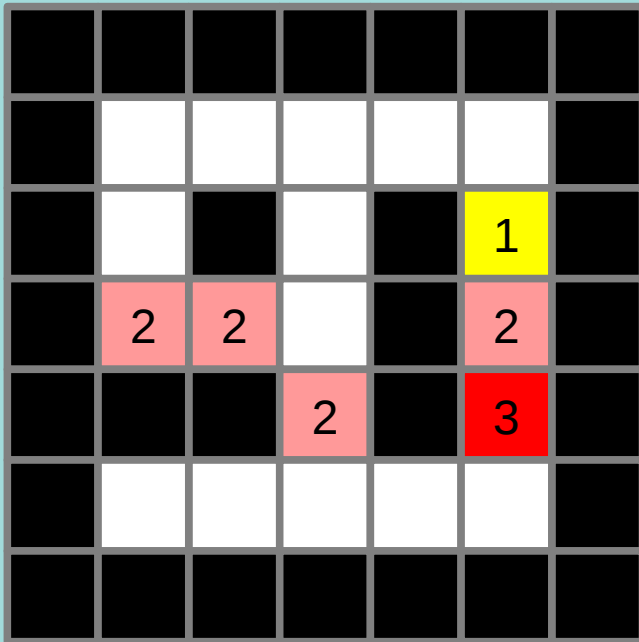


Distances

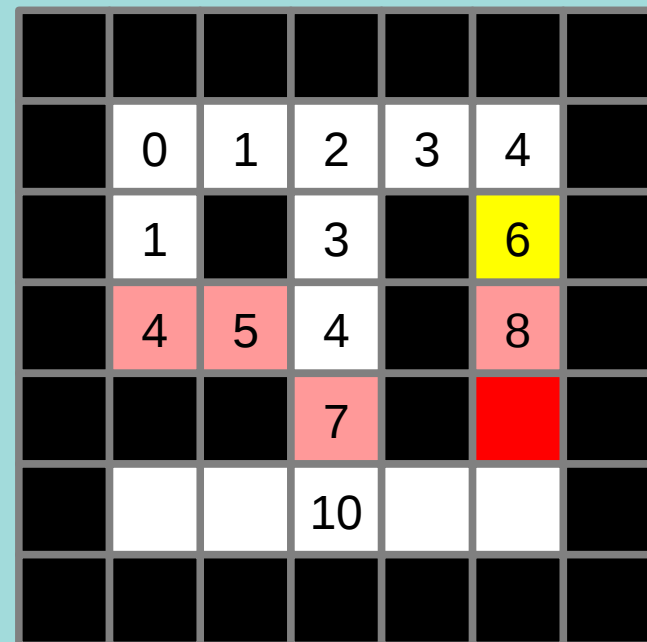
Dijkstra => plus proche d'abord

# Dijkstra

- Calcul des distances et Propagation



Cout = difference altitude

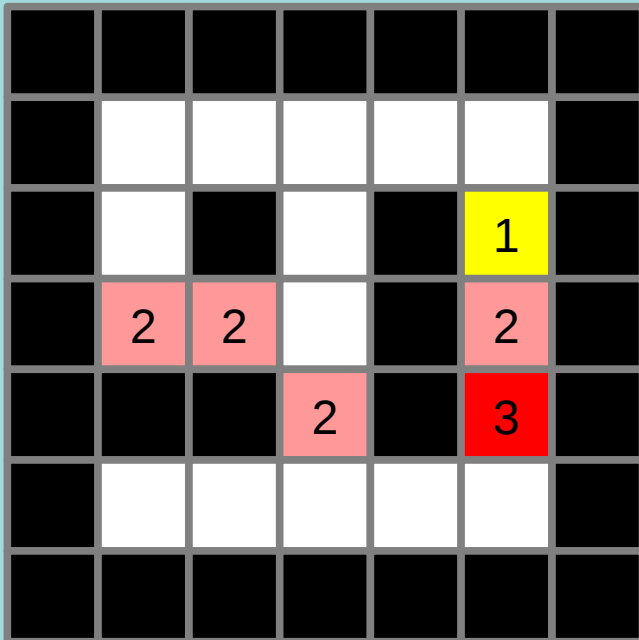


Distances

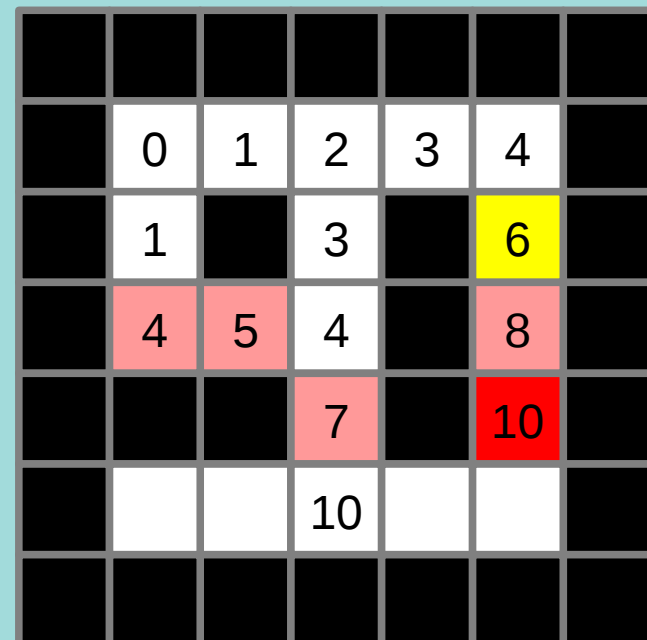
Dijkstra => plus proche d'abord

# Dijkstra

- Calcul des distances et Propagation



Cout = difference altitude

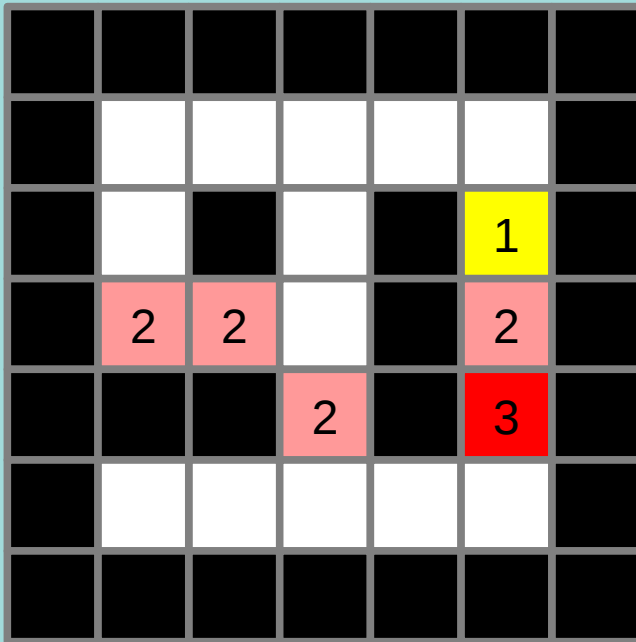


Distances

Dijkstra => plus proche d'abord

# Dijkstra

- Calcul des distances et Propagation



Cost = difference altitude



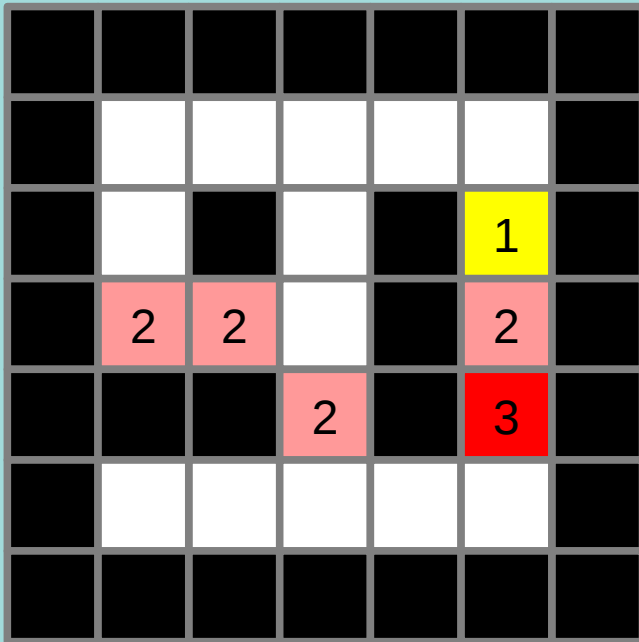
Distances



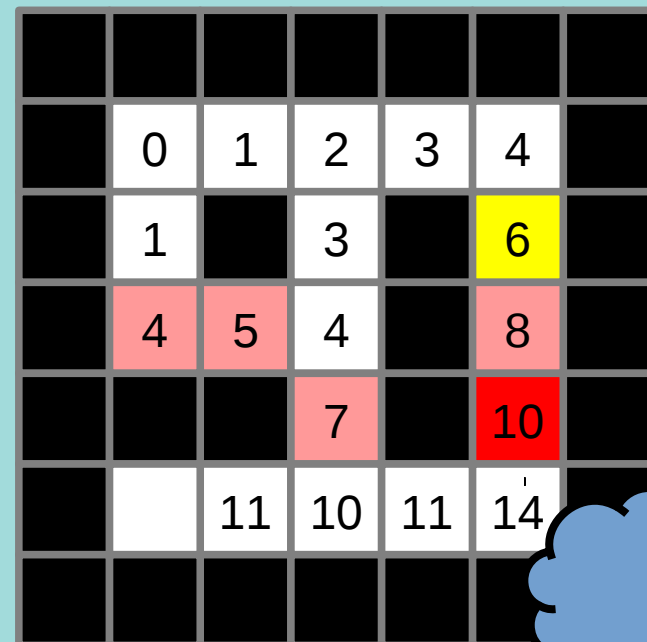
Attention !!!!

# Dijkstra

- Calcul des distances et Propagation



Coût = différence altitude



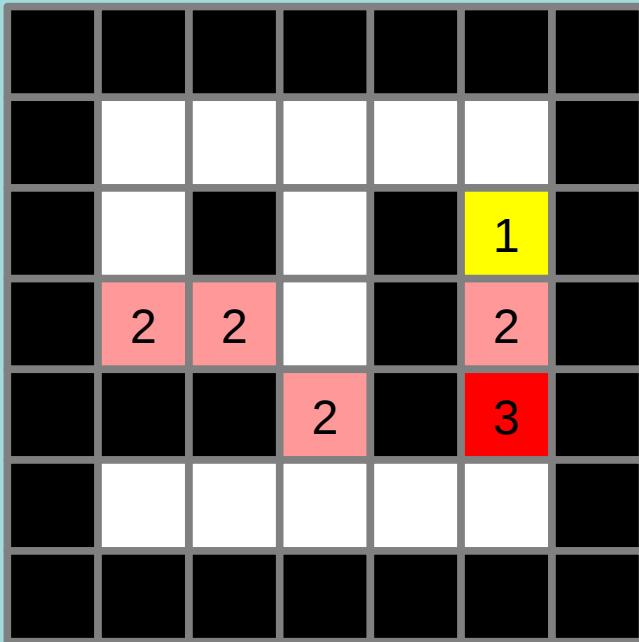
Distances

Attention !!!!

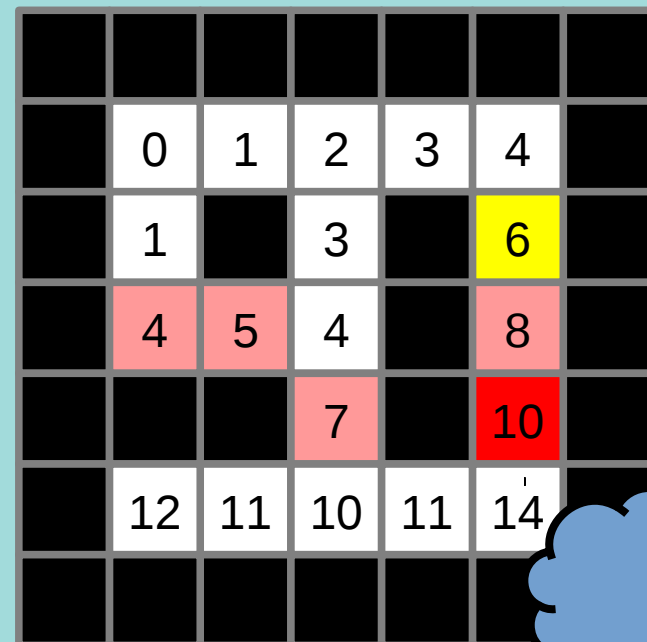


# Dijkstra

- Calcul des distances et Propagation



Coût = différence altitude



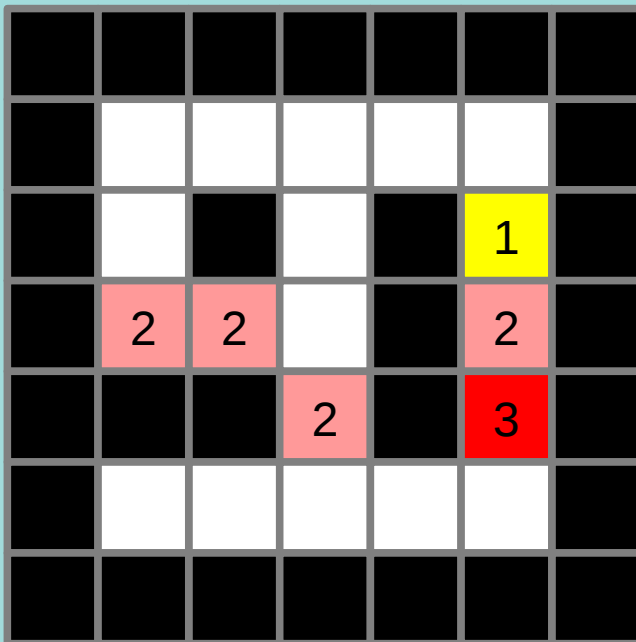
Distances



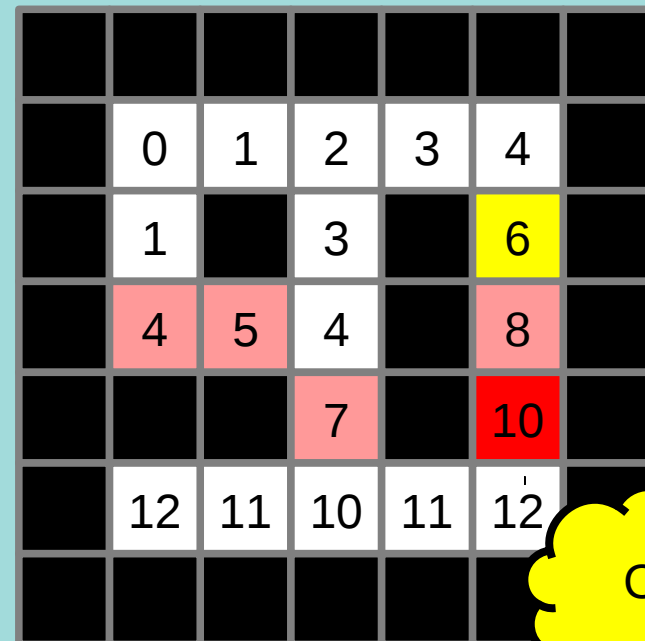
Attention !!!!

# Dijkstra

- Calcul des distances et Propagation



Cout = difference altitude



Distances

OK

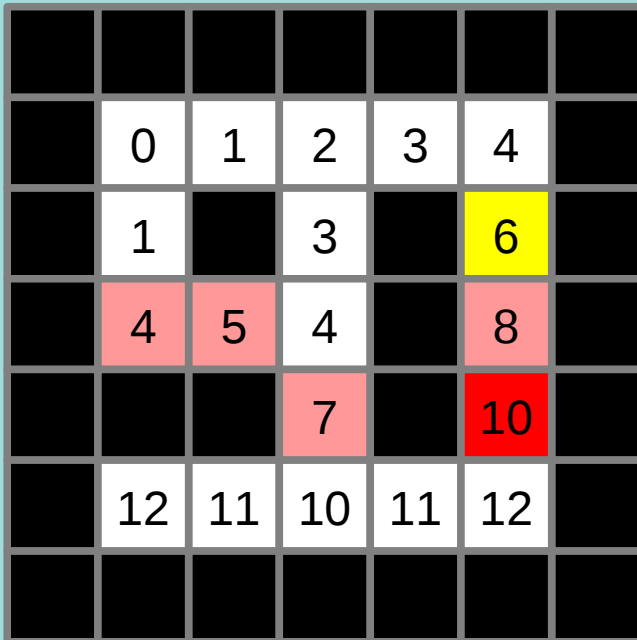
Dijkstra => remplacer si meilleur

# Dijkstra

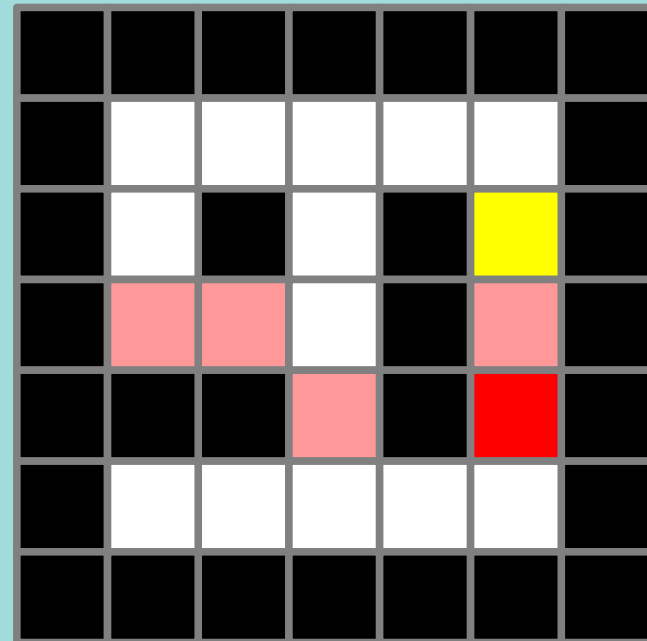
- Chemin

# Dijkstra

- Chemin



Distances

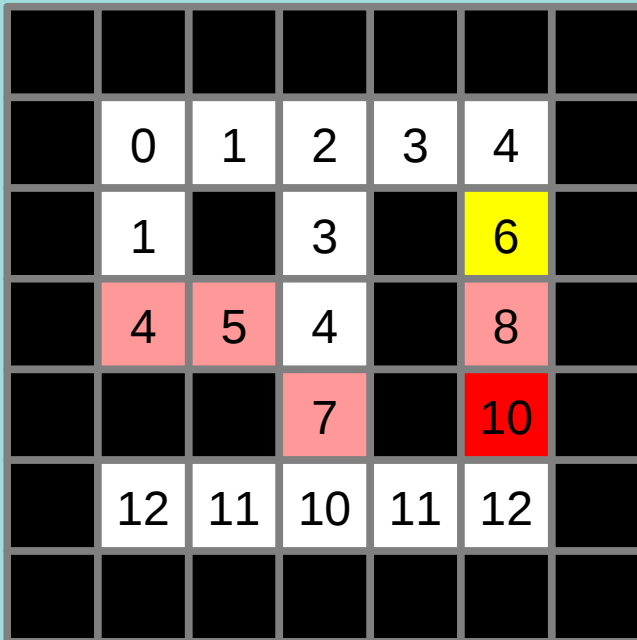


Chemins

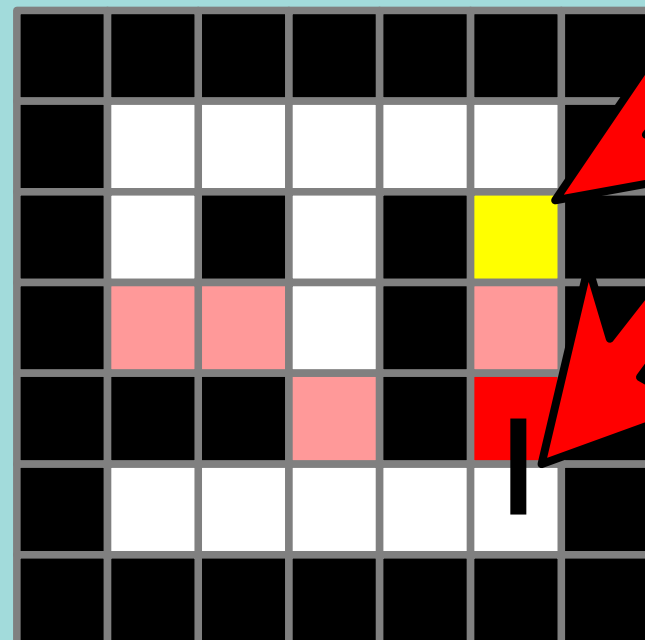
Verifier chemin valide

# Dijkstra

- Chemin



Distances

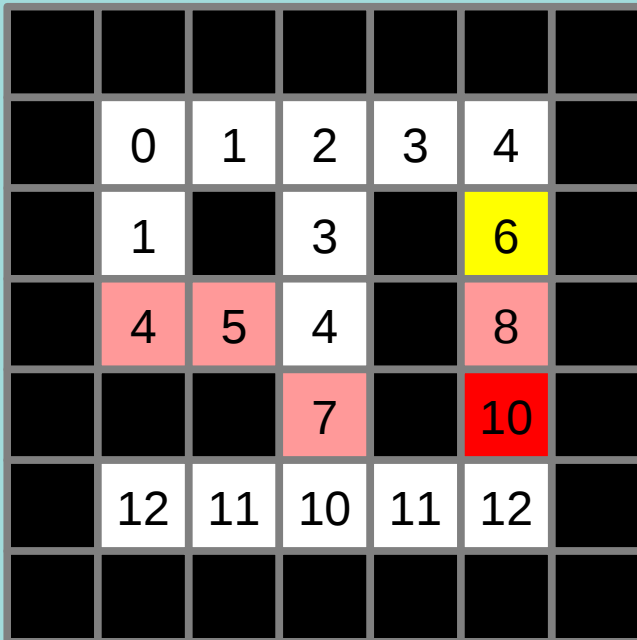


Chemins

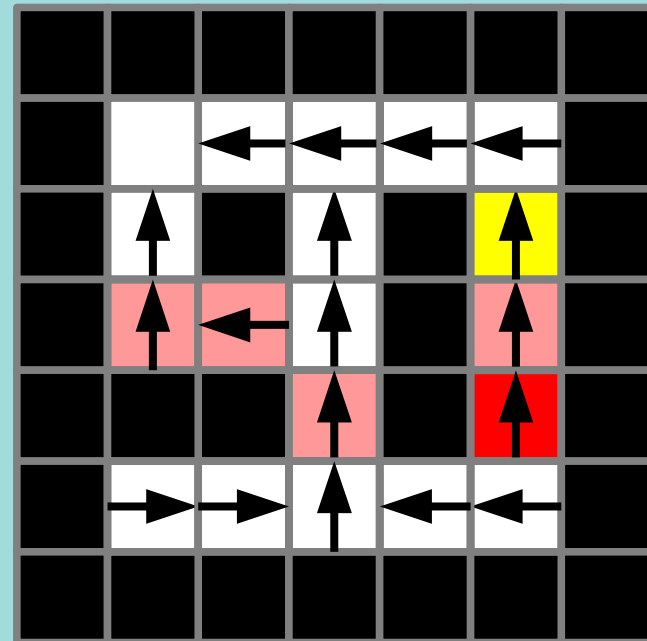
Chemin **non optimal** : car 12 ne vient pas du 10

# Dijkstra

- Chemin



Distances



Chemins

Chemins **optimaux** : verifier bon coût

# Dijkstra

- Algorithme
  - Utiliser liste ouverte pour gérer frontière
  - Développer élément cout le plus petit (tri de la liste)
  - Changer si nouvelle valeur  $<$  valeur présente
- Preuve
  - Comme coûts  $> 0$
  - Quand active une case, on pourra pas mieux
  - Car tous les autres cases, valeur plus grande