Recherche, heuristiques, arbres, jeux

Jill-Jênn Vie

11 janvier 2022

Algorithme de Dijkstra

- Comment ça marche ?
- Quelle complexité ?

Algorithme de Dijkstra

```python
def dijkstra(graph, weight, source=0, target=None):
    n = len(graph)
    prec = [None] * n
    black = [False] * n
    dist = [float('inf')] * n
    dist[source] = 0
    heap = [(0, source)]
    while heap:
        dist_node, node = heappop(heap)
        if not black[node]:
            black[node] = True
            if node == target:
                break
            for neighbor in graph[node]:
                dist_neighbor = (dist_node +
                                 weight[node][neighbor])
                if dist_neighbor < dist[neighbor]:
                    dist[neighbor] = dist_neighbor
                    prec[neighbor] = node
                    heappush(heap, (dist_neighbor, neighbor))
    return dist, prec
```

Best-first search

Algorithme Recherche du meilleur en premier

Mettre la *source* dans la file
tant que la file n'est pas vide **faire**
    Extraire le nœud u ayant la priorité $f(u)$ minimale
    si u est la *cible* **alors renvoyer** dist, prec
    pour tout voisin v du nœud u **faire**
        candidat $\leftarrow$ dist(u) + poids arête w_{uv}
        si c'est un meilleur candidat i.e. candidat $<$ dist(voisin) **alors**
            dist(voisin v) $\leftarrow$ candidat
            prec(voisin v) $\leftarrow$ nœud u
            Ajouter v à la file avec priorité $f(v)$

Valeurs de priorité $f(u)$

- Dijkstra : parcourir par plus courte distance(source, nœud u)
- *Greedy best-first search* : $h(u)$ distance à vol d'oiseau à l'arrivée
- A* : par $dist(u) + h(u)$ croissant.
- Prim (arbre couvrant) : distance à l'arbre en cours

Heuristiques admissibles

Définition

Une heuristique est admissible si elle sous-estime toujours la vraie distance à la cible t : $\forall u \in V, h(u) \leq d(u,t)$.

Lemme

Si l'heuristique est admissible, alors A renvoie un plus court chemin*

Démonstration.

Do it yourself. À vos crayons. Soit C la distance renvoyée.
Supposons qu'il y ait un autre u non traité pour lequel la distance est optimale $C^* < C$.
$f(u) > C > C^*$ sinon u aurait été traité par l'algorithme.
$f(u) = dist(u) + h(u)$ (def)
$f(u) = dist^*(u) + h(u)$ (car u est sur un chemin optimal)
$f(u) \leq dist^*(u) + d(u,t) = C^*$ (car h est admissible).
Contradiction. □

Heuristiques monotones (Russell et Norvig, 2020, p. 88)

Définition

L'heuristique est dite monotone si $\forall n, n' \in E, h(n) \leq w_{nn'} + h(n')$

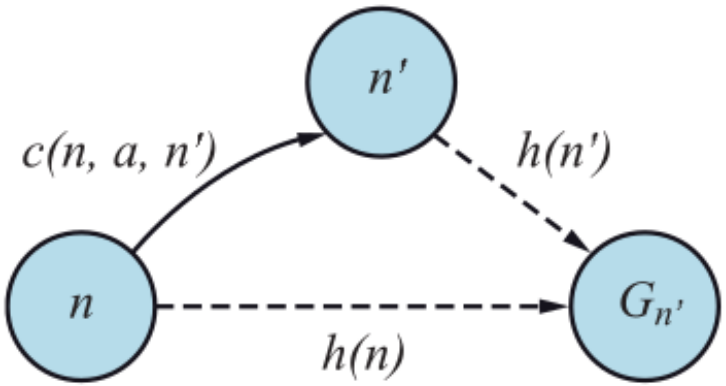


Figure 3.19 Triangle inequality: If the heuristic h is **consistent**, then the single number $h(n)$ will be less than the sum of the cost $c(n,a,a')$ of the action from n to n' plus the heuristic estimate $h(n')$.

Lemme

Si l'heuristique est monotone ($\Rightarrow$ admissible) alors de plus on ne visite chaque sommet qu'une fois.

Démonstration.

A* devient équivalent à Dijkstra avec les coûts réduits $w'_{nn'} = w_{nn'} + h(n') - h(n)$.

□

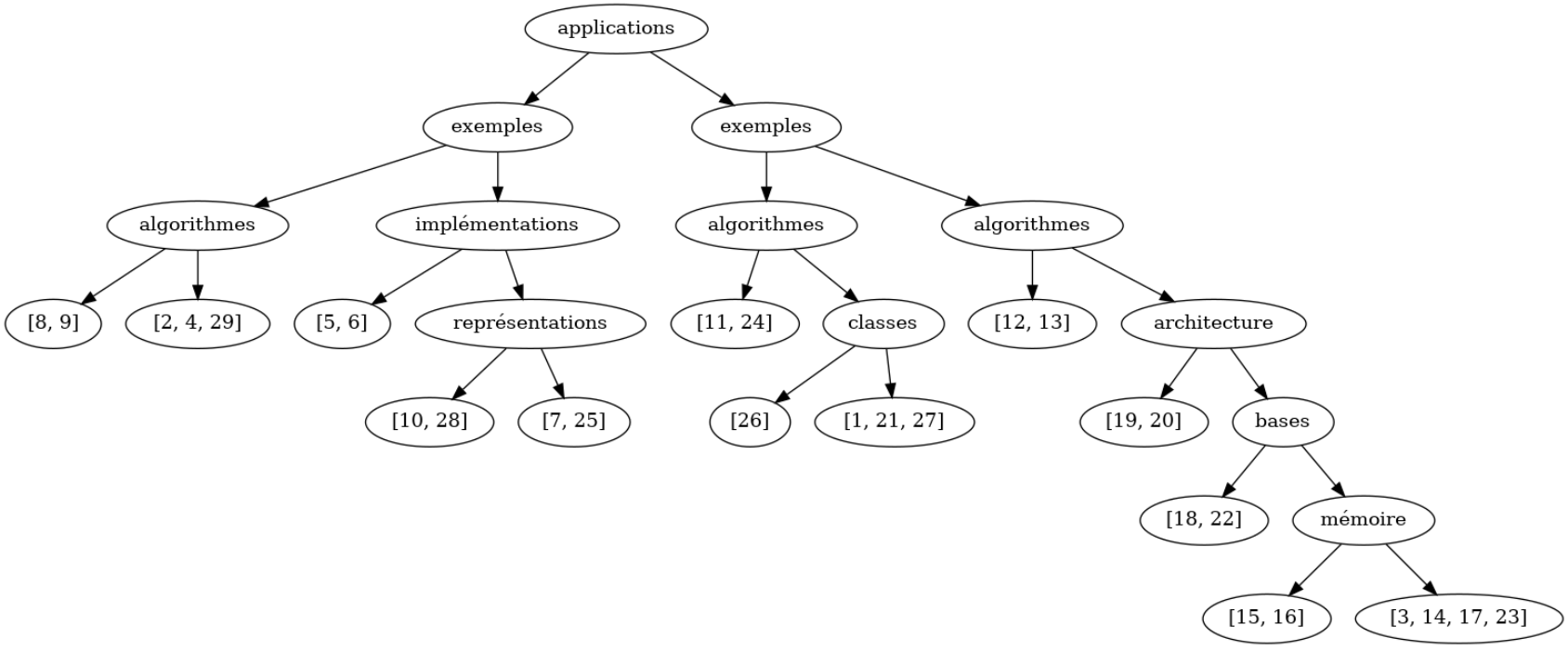
Complexité de A*

$O(|E| \log |V|)$ avec un tas binaire

$O(|E| + |V| \log |V|)$ avec un tas de Fibonacci

On voit beaucoup $O(b^d)$ mais c'est faux, ou alors la complexité est seulement décrite en nombre d'états visités.

Arbres de décision

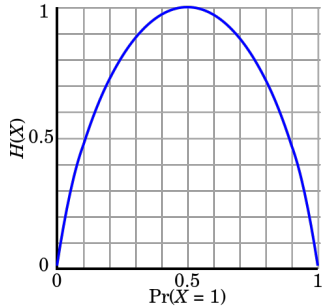


Jeu de données : intitulé des leçons de l'agrégation d'informatique.
Matrice de taille 29 leçons $\times$ 105 mots.

Algorithme ID3 (Russell et Norvig, 2020)

Entropie

$$H(S) = \sum_{c \in \mathcal{Y}} -p(c) \log p(c)$$



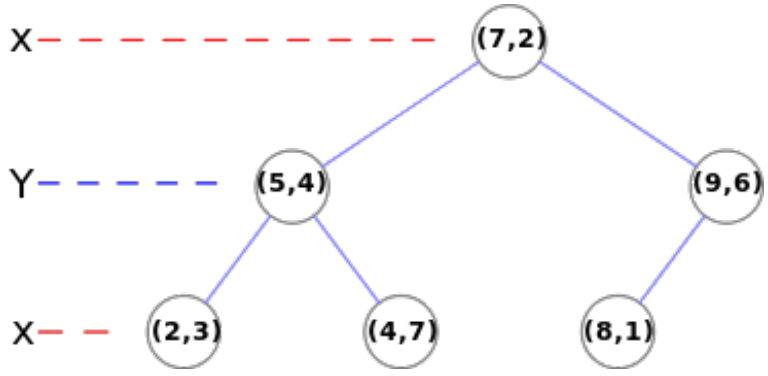
Gain d'information

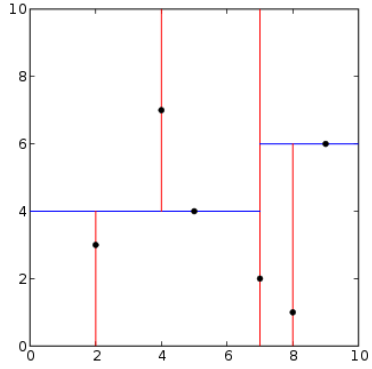
$$IG(S, A) = H(S) - \sum_{S_i} p(A = a_i) H(S_i) = H(S) - H(S|A)$$

où $\bigcup_i S_i = S$ en fonction de l'attribut $A = a_i$

Arbre k-dimensionnel (k-d *tree*)

Généralisation de l'arbre binaire de recherche à plusieurs dimensions



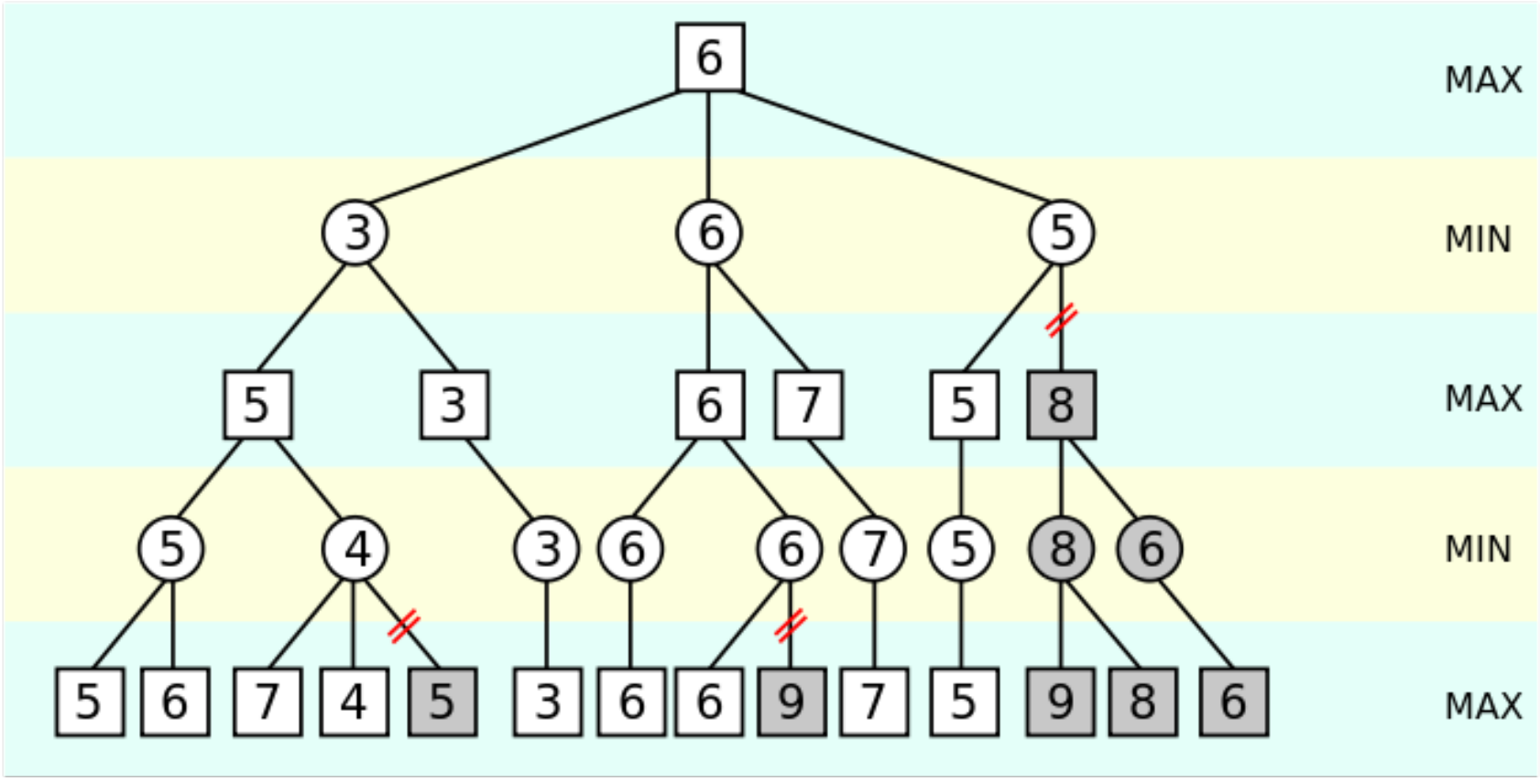


Critère : la médiane selon la dimension en cours (*depth* mod k)

- $O(n \log^2 n)$ pour le construire
- $O(n \log n)$ avec médiane des médianes
- $O(\log n)$ pour trouver un point
- $O(\log n)$ pour trouver un plus proche voisin

  (Friedman, Bentley et Finkel, 1977)
- Utile si $n > 2^k$

Jeux sur des arbres : minimax et élagage alpha-bêta




Friedman, Jerome H, Jon Louis Bentley et Raphael Ari Finkel (1977). “An algorithm for finding best matches in logarithmic expected time”. In : *ACM Transactions on Mathematical Software (TOMS)* 3.3, p. 209-226.


Russell, Stuart et Peter Norvig (2020). *Artificial Intelligence: A Modern Approach*.