Bing

Voici un exemple complexe d'algorithme sur l'approche descendante :

Problème : Trouver le plus court chemin entre deux villes dans un réseau routier.

Objectif : Donner la liste des villes à traverser et la distance totale du trajet.

Données : Le réseau routier est représenté par un graphe, où les sommets sont les villes et les arêtes sont les routes, pondérées par leur longueur.

Algorithme :

1. Initialiser un ensemble vide S qui contiendra les villes visitées, et un ensemble Q qui contiendra les villes à visiter, avec la ville de départ comme seule élément.
2. Initialiser un tableau D qui contiendra les distances minimales entre la ville de départ et chaque autre ville, avec la valeur 0 pour la ville de départ et l'infini pour les autres.
3. Initialiser un tableau P qui contiendra le prédécesseur de chaque ville dans le plus court chemin, avec la valeur null pour la ville de départ et les autres.
4. Tant que Q n'est pas vide, faire :
   1. Sélectionner dans Q la ville u qui a la plus petite distance dans D, et l'ajouter à S.
   2. Si u est la ville d'arrivée, alors arrêter l'algorithme.
   3. Sinon, pour chaque ville v voisine de u qui n'est pas dans S, faire :
      1. Calculer la distance entre u et v par la route qui les relie, et l'ajouter à la distance entre la ville de départ et u dans D.
      2. Si cette somme est inférieure à la distance entre la ville de départ et v dans D, alors mettre à jour D avec cette somme, et mettre à jour P avec u comme prédécesseur de v.
   4. Calculer la distance entre u et v par la route qui les relie, et l'ajouter à la distance entre la ville de départ et u dans D.
   5. Si cette somme est inférieure à la distance entre la ville de départ et v dans D, alors mettre à jour D avec cette somme, et mettre à jour P avec u comme prédécesseur de v.
5. Sélectionner dans Q la ville u qui a la plus petite distance dans D, et l'ajouter à S.
6. Si u est la ville d'arrivée, alors arrêter l'algorithme.
7. Sinon, pour chaque ville v voisine de u qui n'est pas dans S, faire :
   1. Calculer la distance entre u et v par la route qui les relie, et l'ajouter à la distance entre la ville de départ et u dans D.
   2. Si cette somme est inférieure à la distance entre la ville de départ et v dans D, alors mettre à jour D avec cette somme, et mettre à jour P avec u comme prédécesseur de v.
8. Calculer la distance entre u et v par la route qui les relie, et l'ajouter à la distance entre la ville de départ et u dans D.
9. Si cette somme est inférieure à la distance entre la ville de départ et v dans D, alors mettre à jour D avec cette somme, et mettre à jour P avec u comme prédécesseur de v.
10. Construire le plus court chemin en partant de la ville d'arrivée et en remontant les prédécesseurs dans P jusqu'à la ville de départ, et inverser l'ordre des villes.
11. Afficher le plus court chemin et la distance correspondante.

Cet algorithme est connu sous le nom d'algorithme de Dijkstra. Il utilise l'approche descendante car il part du problème global (trouver le plus court chemin entre deux villes) et le décompose en sous-problèmes plus simples (trouver la ville la plus proche, mettre à jour les distances, construire le chemin). Il utilise également des structures fondamentales comme les boucles, les conditions, les tableaux et les ensembles. Il pourrait être implémenté en utilisant la programmation modulaire, en définissant des fonctions pour chaque étape de l'algorithme.