Recherche Operationnelle

Vincent Gripon

École Nationale Supérieure d'Ingénieurs de Brest

18 mai 2010

Voyageur de commerce

Formulation du problème

- Un commerçant itinérant souhaite se rendre dans toutes les villes d'une région donnée puis revenir à son point de départ. Le problème du voyageur de commerce est le suivant : dans quel ordre les visiter de manière à réduire le temps total de trajet ?
- Exemple : Optimiser une chaîne de montage...

Exemple sur le labyrinthe

- Objectif : récupérer des surprises dans le labyrinthe.
- Modélisation : un nouveau graphe est défini tel que :
  - Les noeuds correspondent aux surprises,
  - Les arrêtes sont pondérées par le longueur d'un chemin minimal reliant les surprises dans le labyrinthe.
- Remarques :
  - Le graphe obtenu est complet et respecte l'inégalité triangulaire.

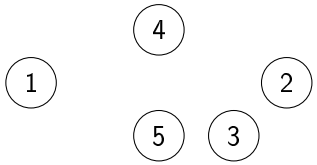
Première approche

Méthodes naïves

- Une première idée est d'utiliser un algorithme glouton :
  - Un tel algorithme se base sur une heuristique (= algorithme qui prend une décision raisonable bien que non nécessairement optimale),
  - Exemple : choisir à chaque étape le voisin le plus proche,
- Une approche duale est de calculer la valeur de tous les chemins, puis de retenir le meilleur : démarche exhaustive.

Problèmes

- L'algorithme glouton n'est pas exact :
- Le calcul de tous les chemins est bien trop couteux ($(n-1)!$) pour être réalisable en pratique.



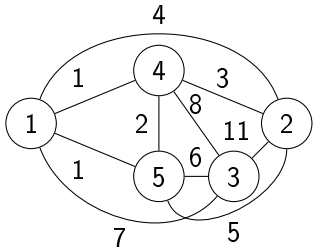
Première approche

Méthodes naïves

- Une première idée est d'utiliser un algorithme glouton :
  - Un tel algorithme se base sur une heuristique (= algorithme qui prend une décision raisonable bien que non nécessairement optimale),
  - Exemple : choisir à chaque étape le voisin le plus proche,
- Une approche duale est de calculer la valeur de tous les chemins, puis de retenir le meilleur : démarche exhaustive.

Problèmes

- L'algorithme glouton n'est pas exact :
- Le calcul de tous les chemins est bien trop couteux ($(n-1)!$) pour être réalisable en pratique.



Et si on était plus malin ?

À partir de la démarche exhaustive

- La démarche exhaustive peut être réalisée soit à partir d'un parcours en largeur ou en profondeur,
- Si une valeur possible de chemin est déjà connue, alors l'algorithme peut être optimisé en s'arrêtant lors de la visite des autres chemins dès que l'on dépasse cette valeur.

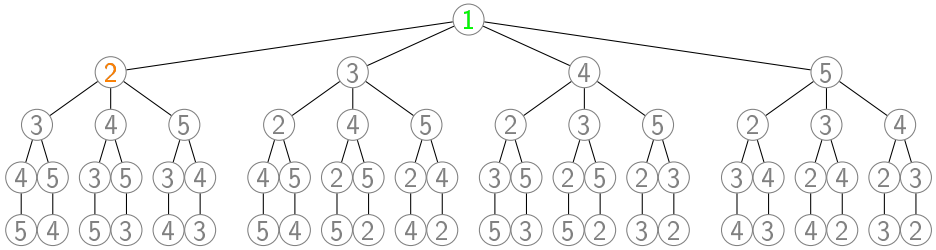
Profondeur ou largeur ?

- L'estimation du chemin optimal est faite de façon naturelle par le parcours en profondeur,
- L'utilisation d'un parcours en largeur évite de compléter trop de chemins,
- Deux possibilités : affiner cette borne en visitant des chemins, ou en calculant des bornes à la volée sur les chemins.

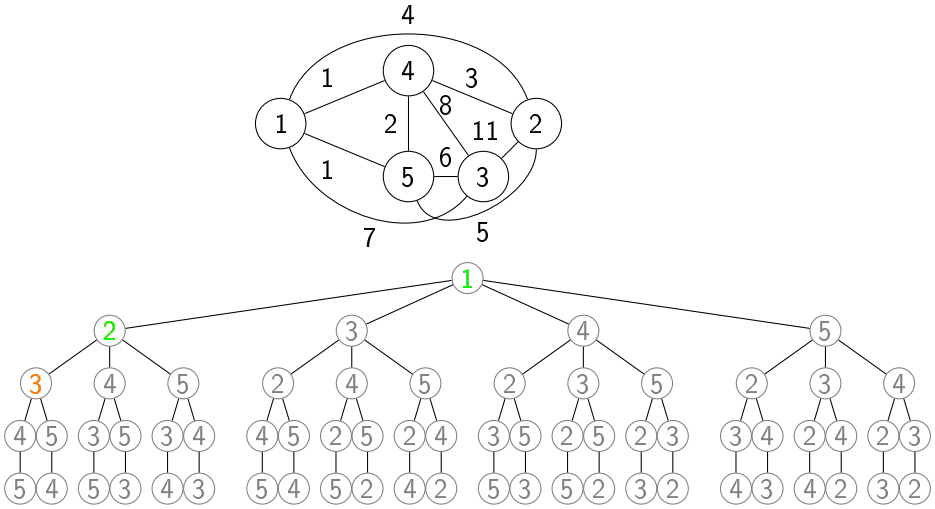
Backtracking

Backtracking

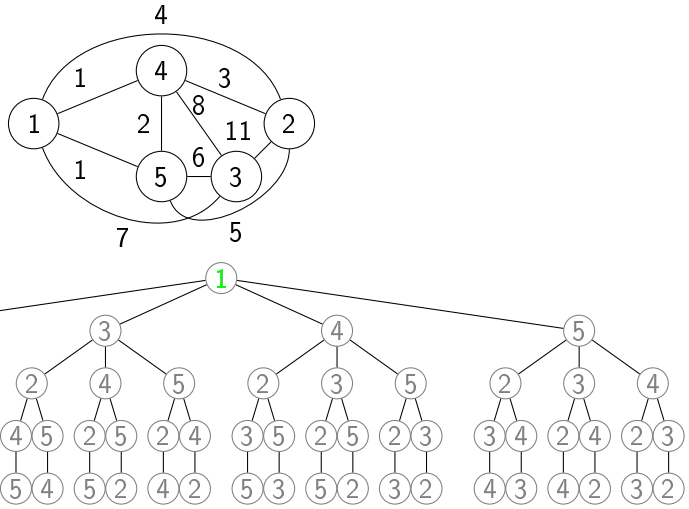
- Parcours en profondeur modifié :
- Dès que le poids d'un chemin dépasse la borne, l'arbre est coupé à cet endroit,
- Quand une branche est finie, la borne est mise à jour.



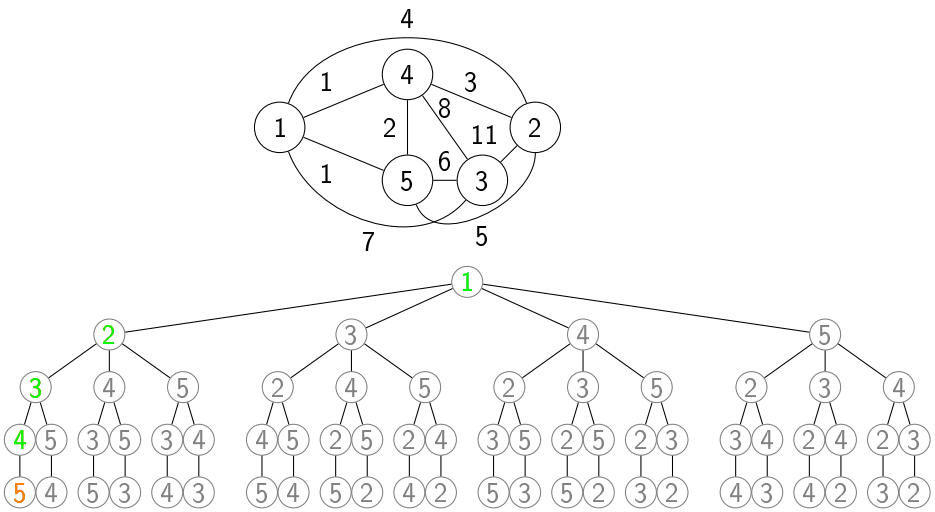
Backtracking



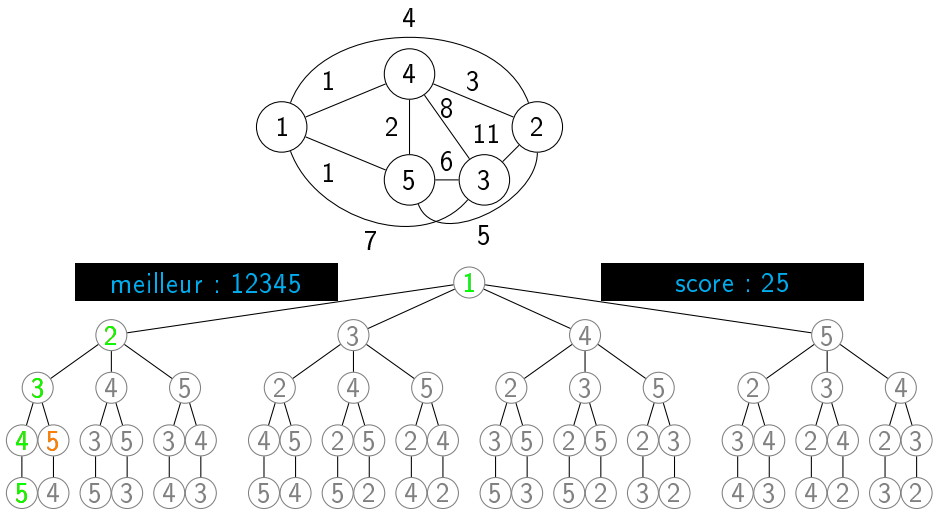
Backtracking



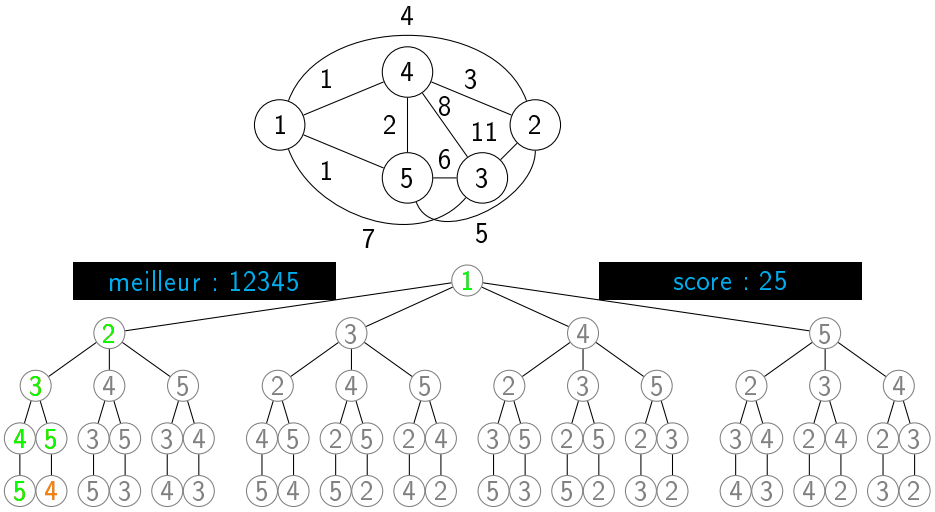
Backtracking



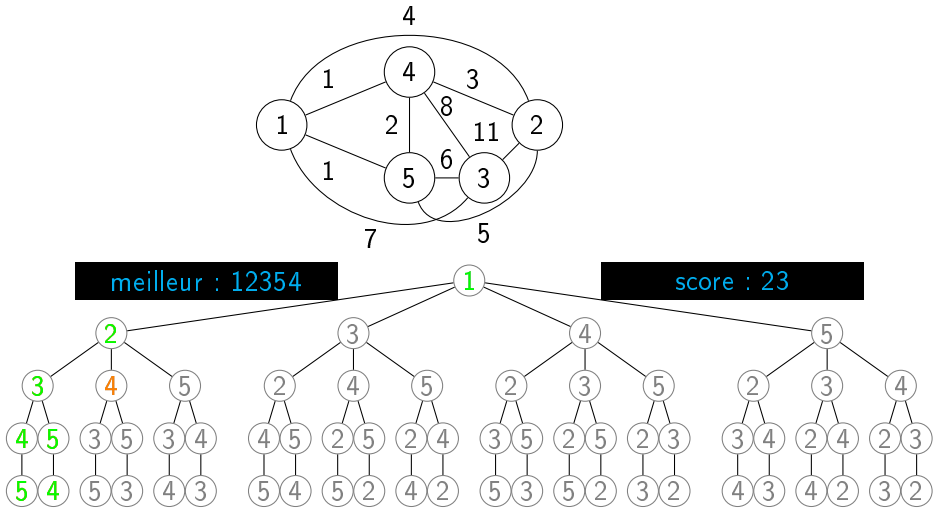
Backtracking



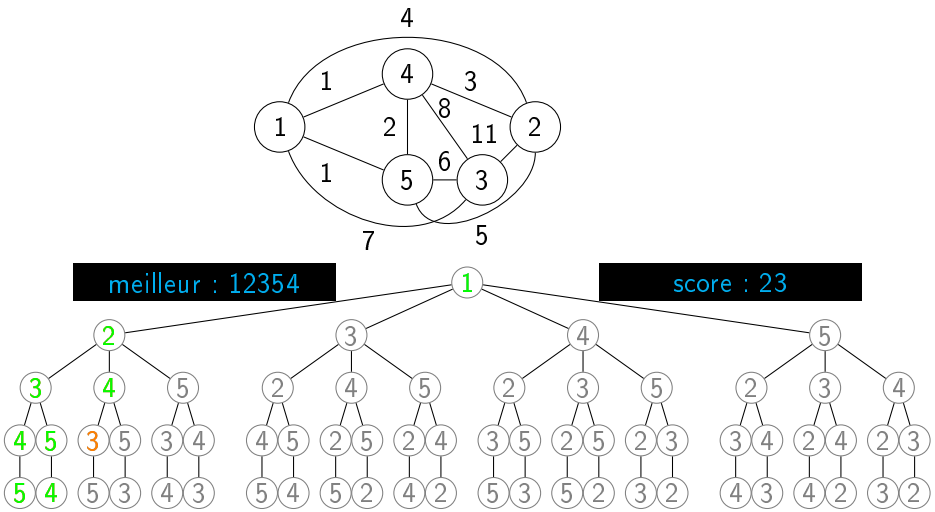
Backtracking



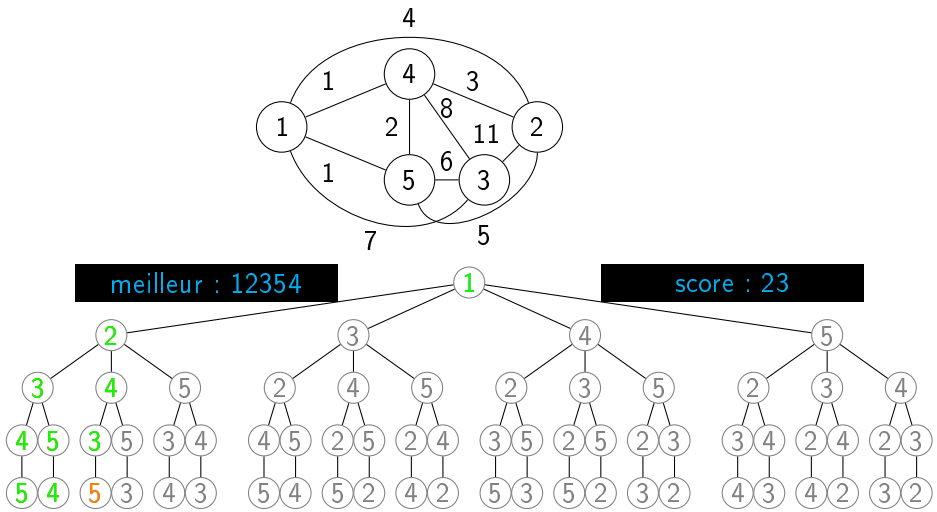
Backtracking



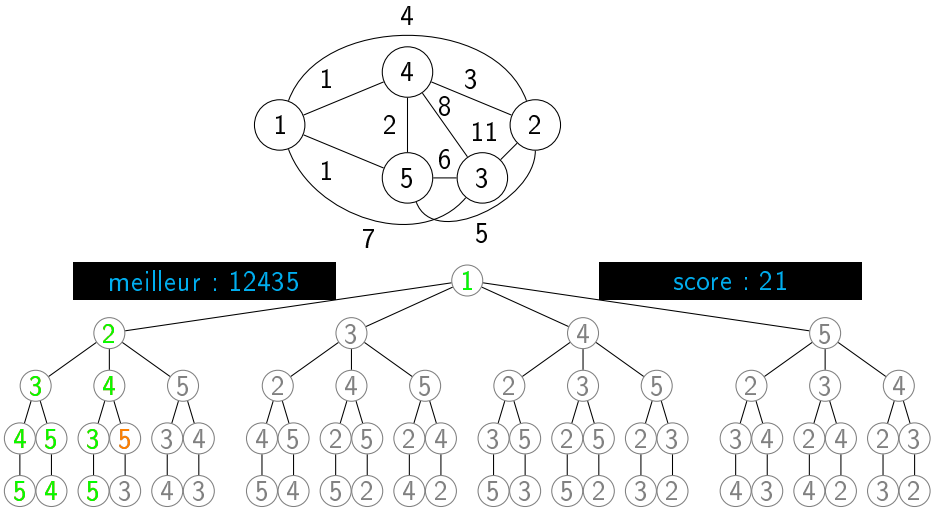
Backtracking



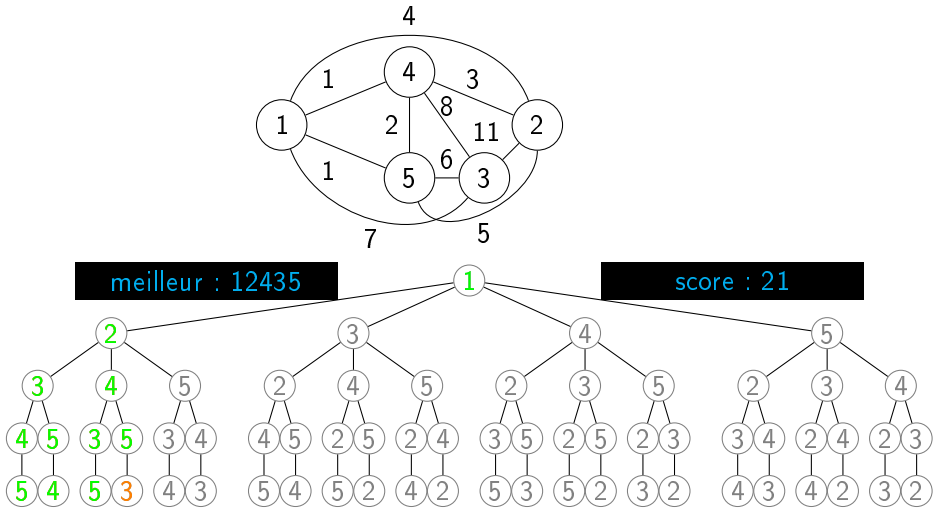
Backtracking



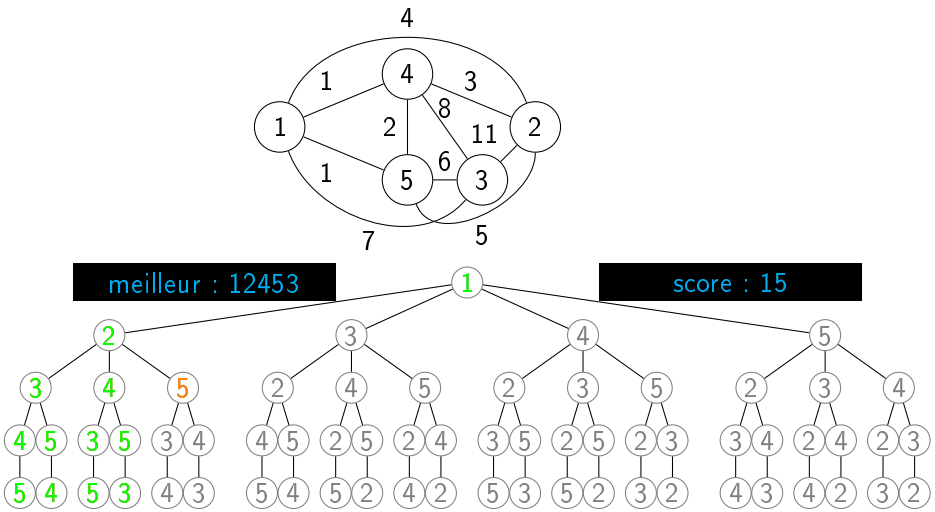
Backtracking



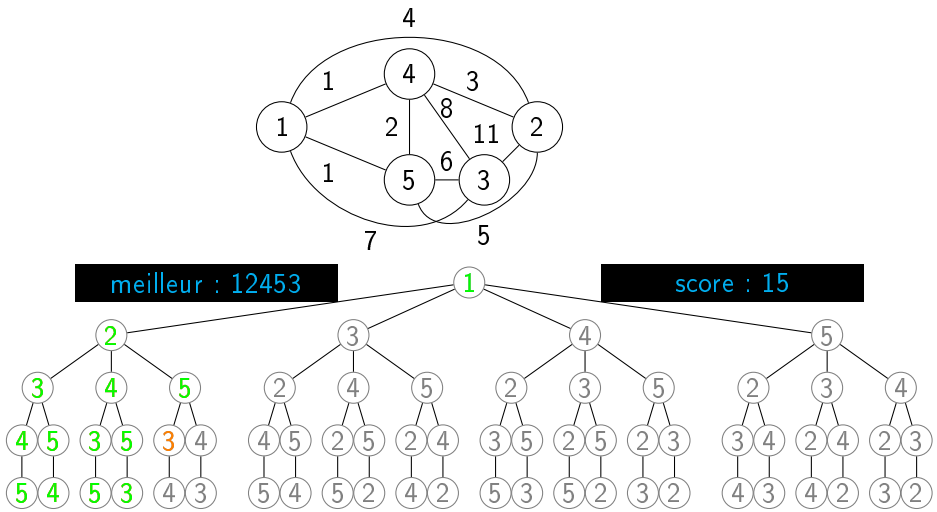
Backtracking



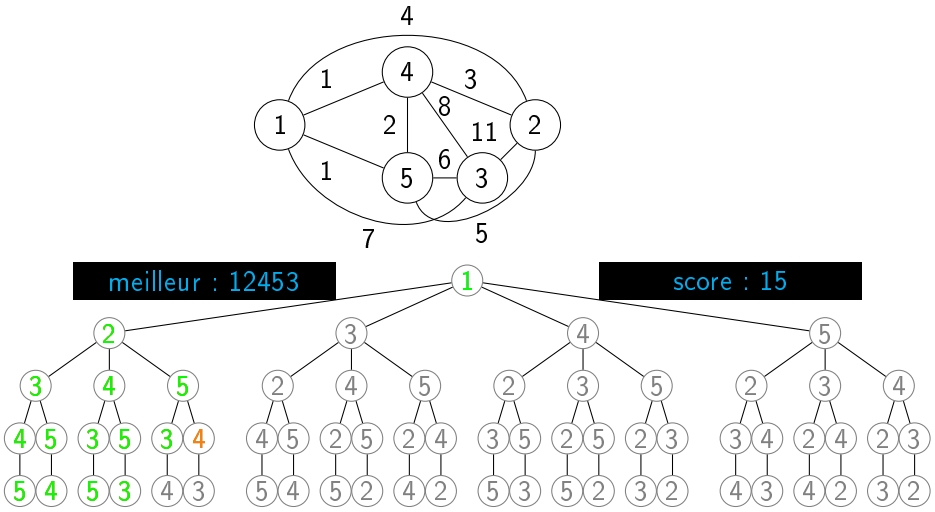
Backtracking



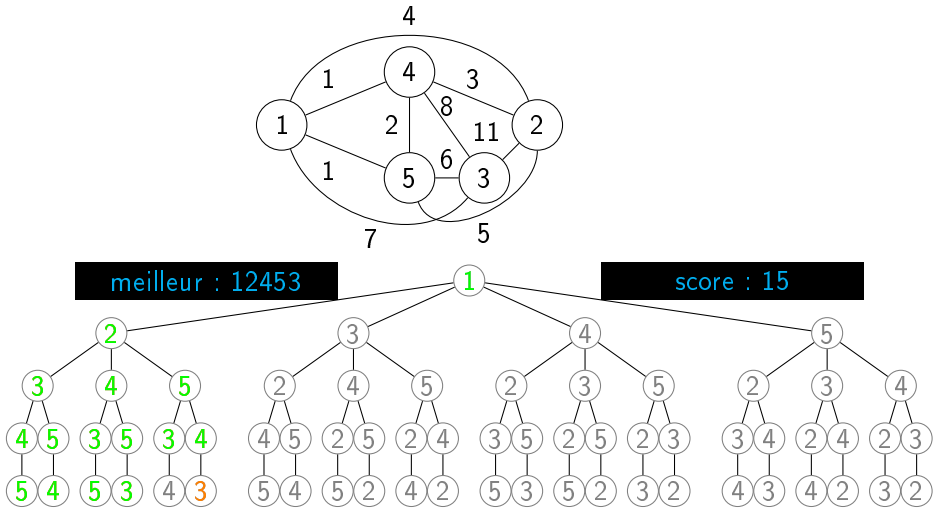
Backtracking



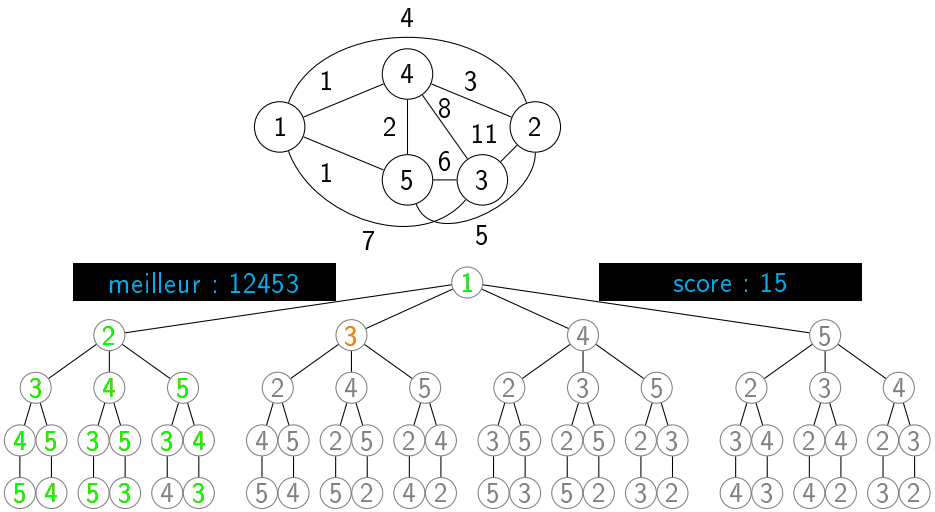
Backtracking



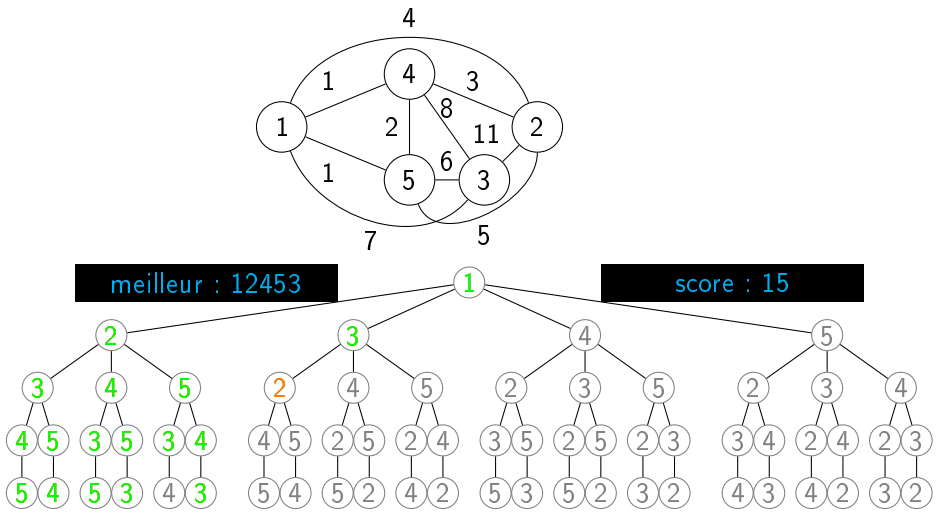
Backtracking



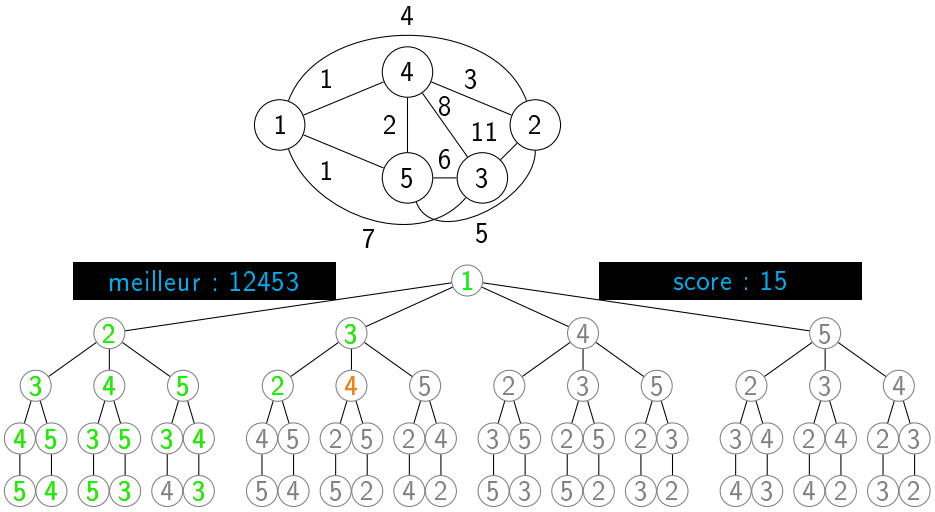
Backtracking



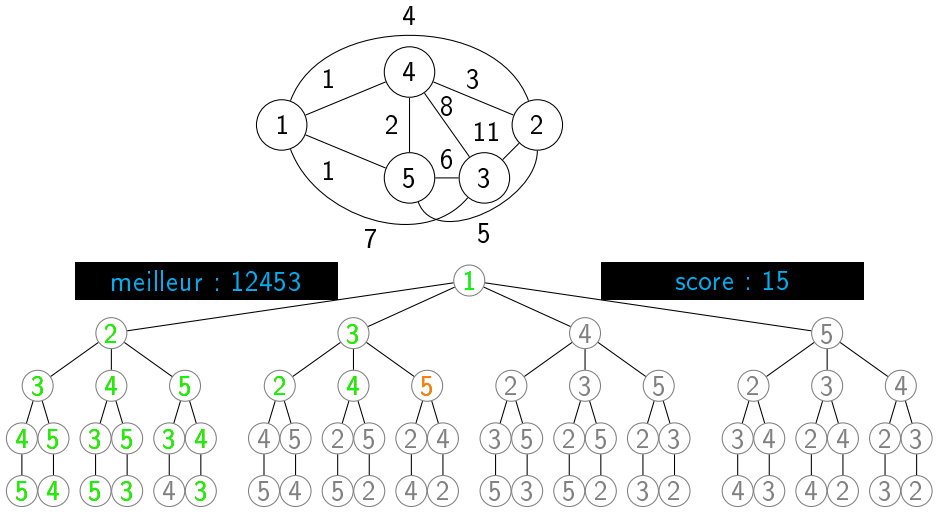
Backtracking



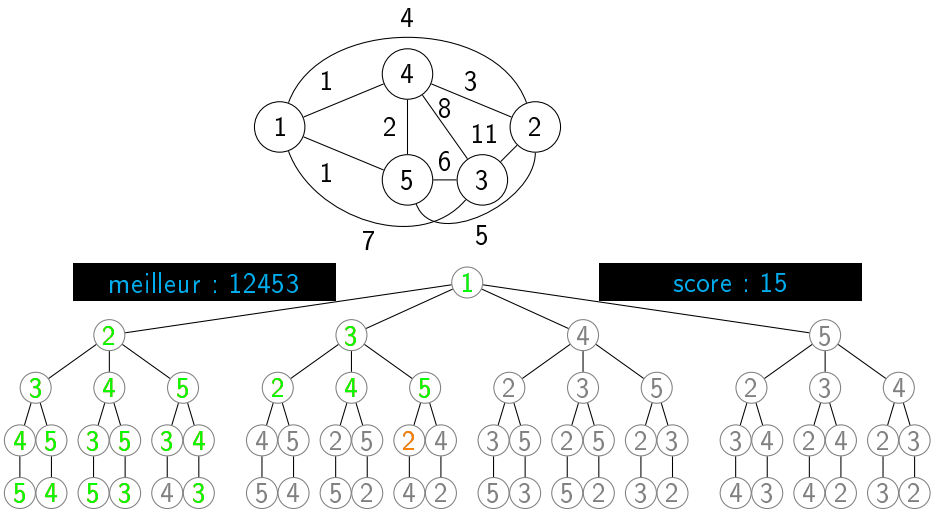
Backtracking



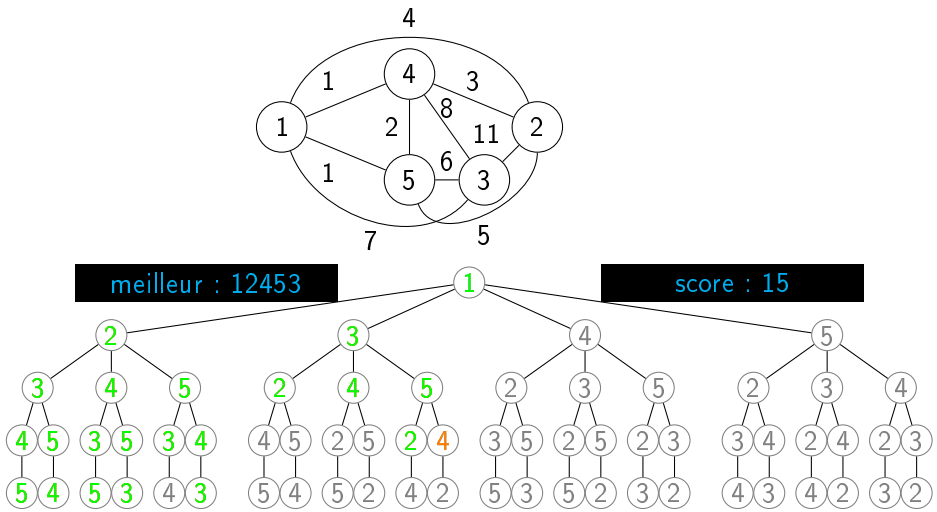
Backtracking



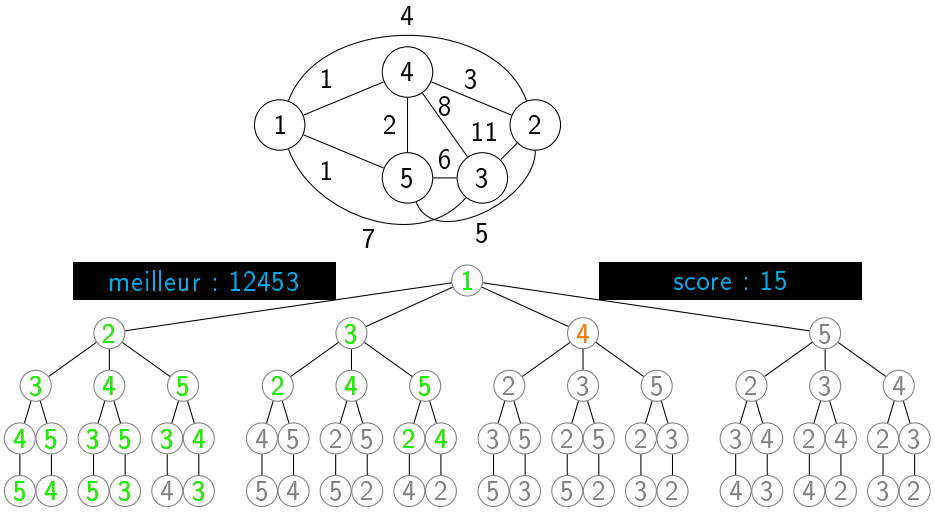
Backtracking



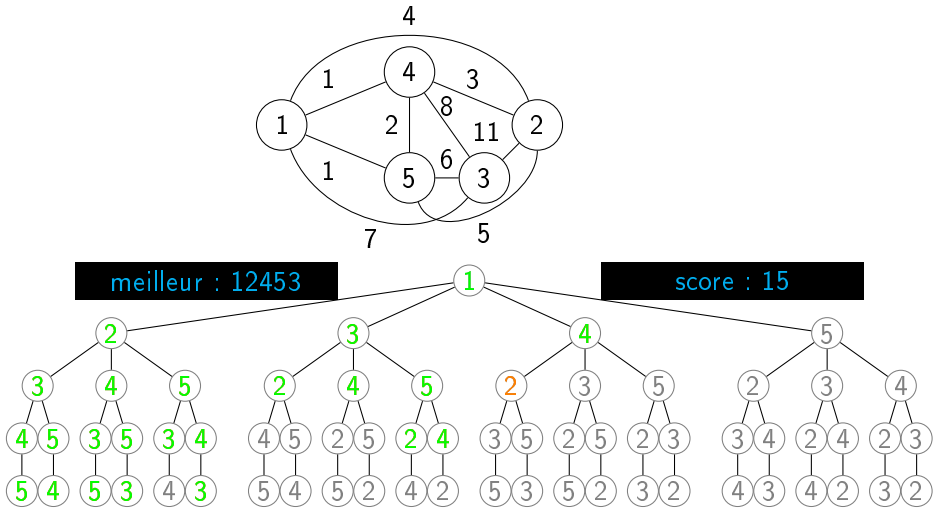
Backtracking



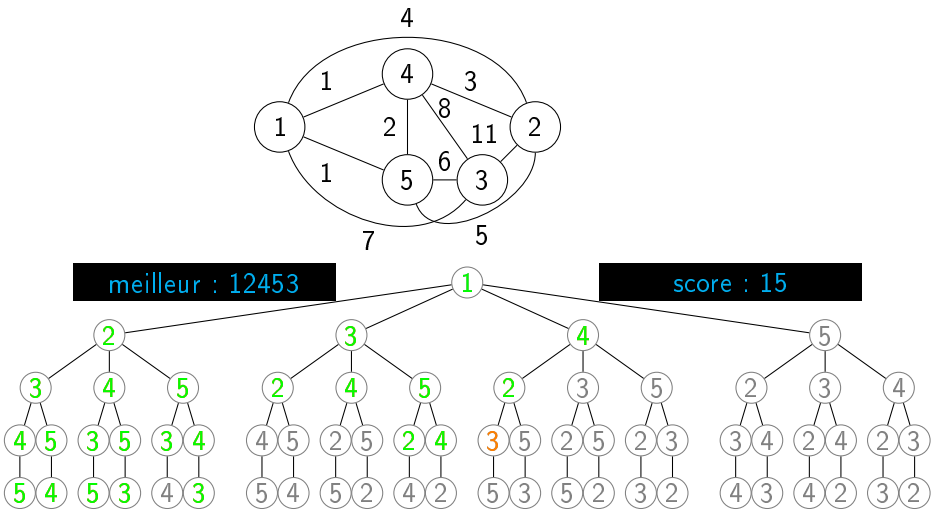
Backtracking



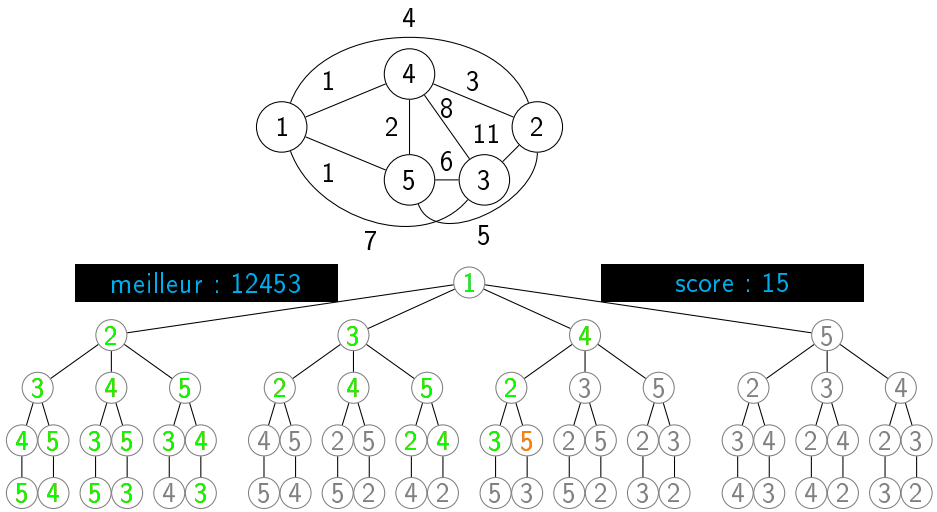
Backtracking



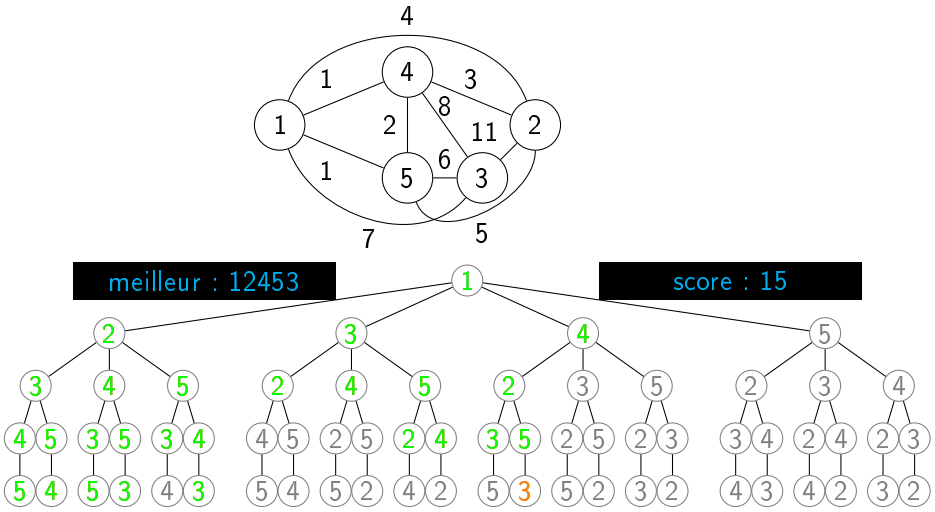
Backtracking



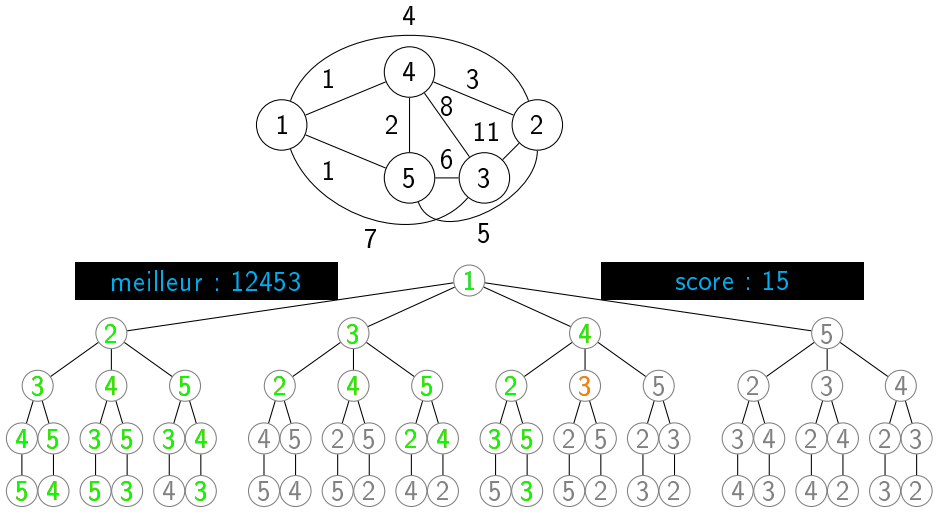
Backtracking



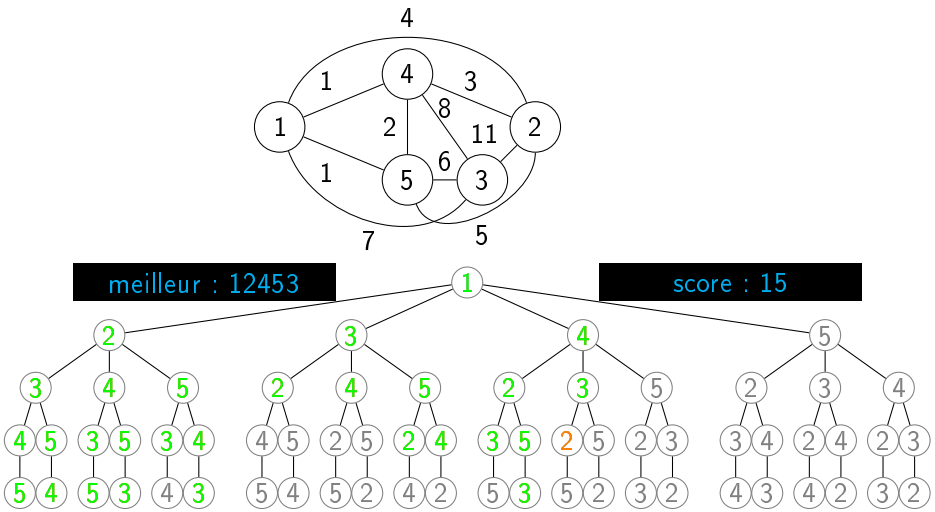
Backtracking



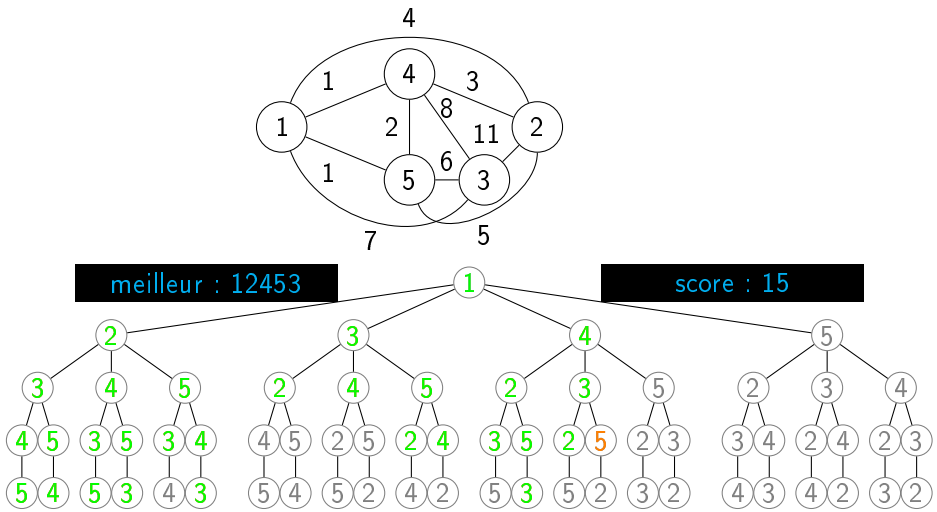
Backtracking



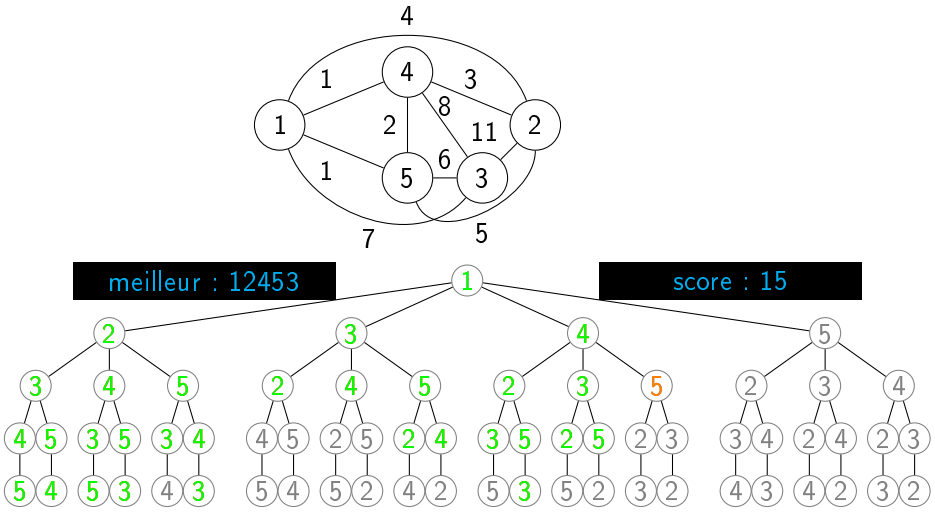
Backtracking



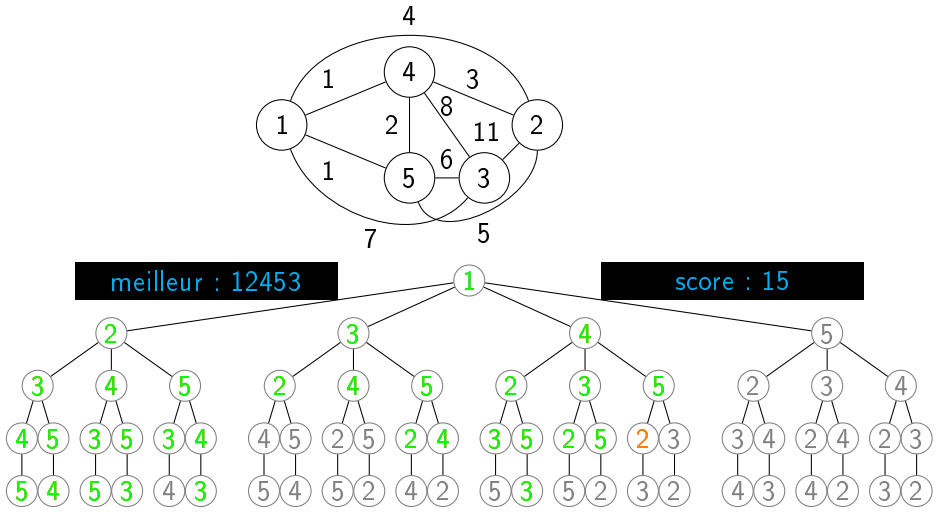
Backtracking



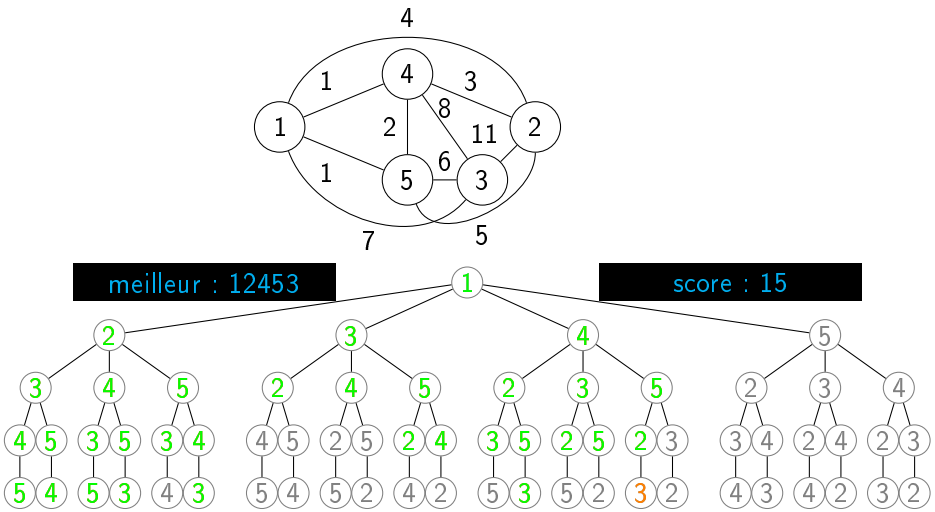
Backtracking



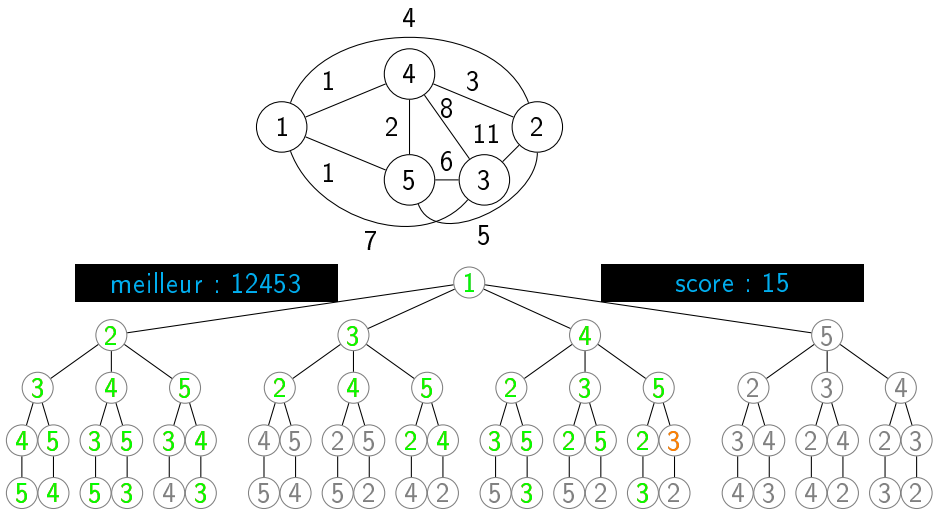
Backtracking



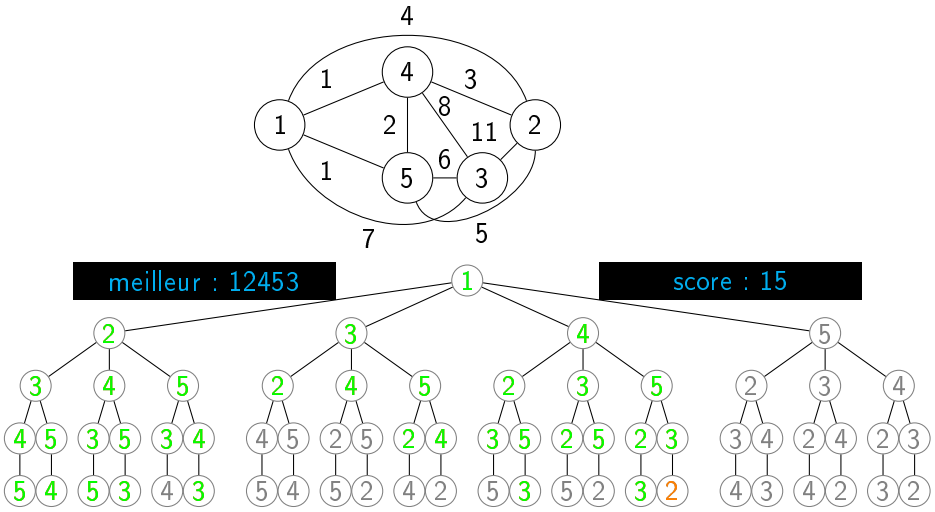
Backtracking



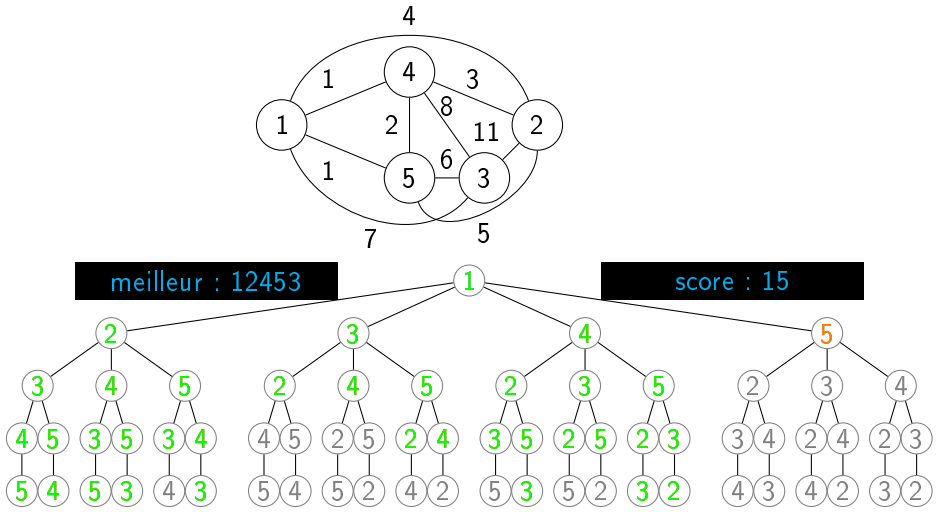
Backtracking



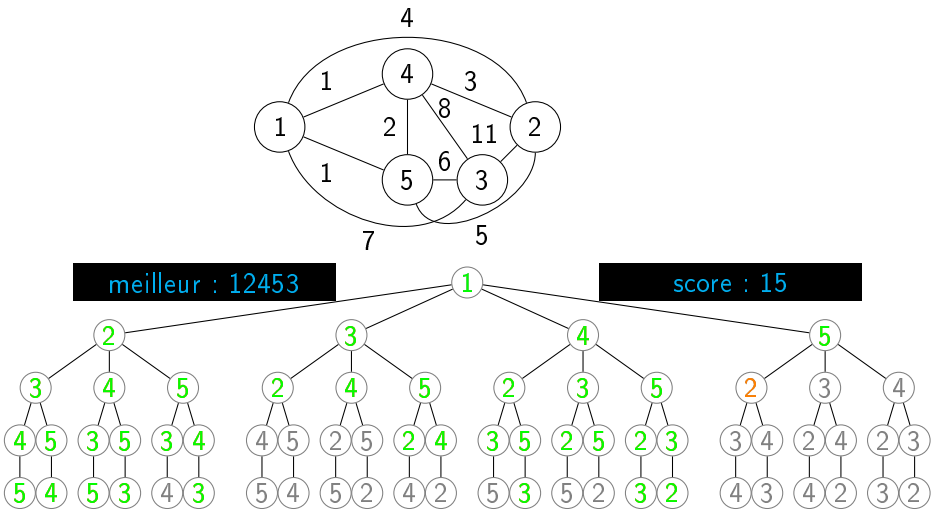
Backtracking



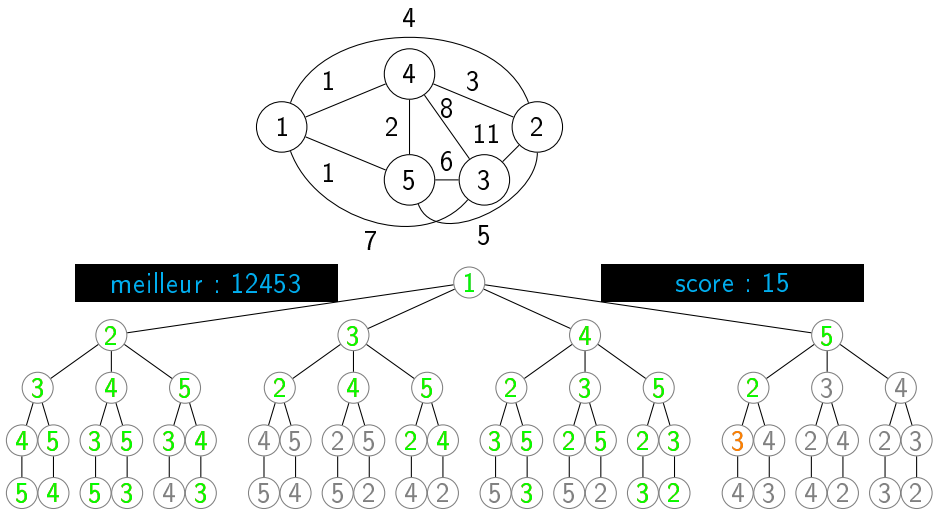
Backtracking



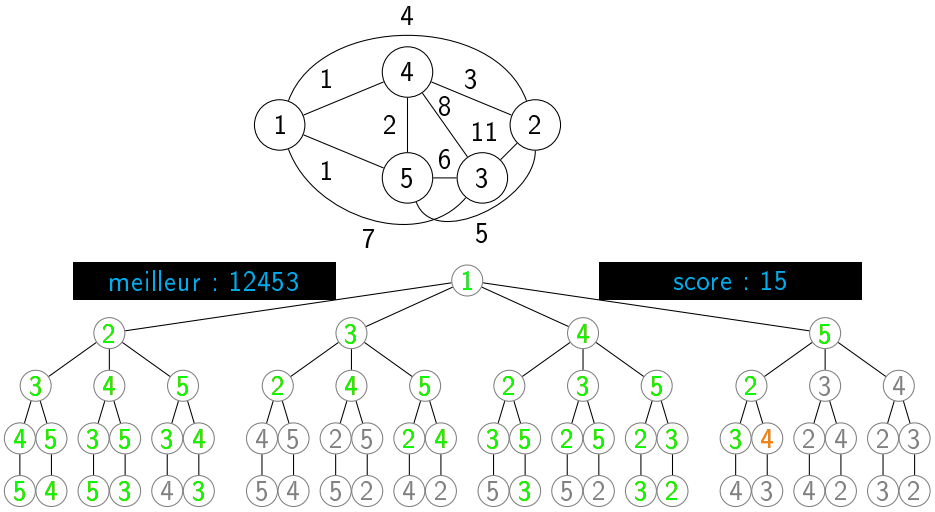
Backtracking



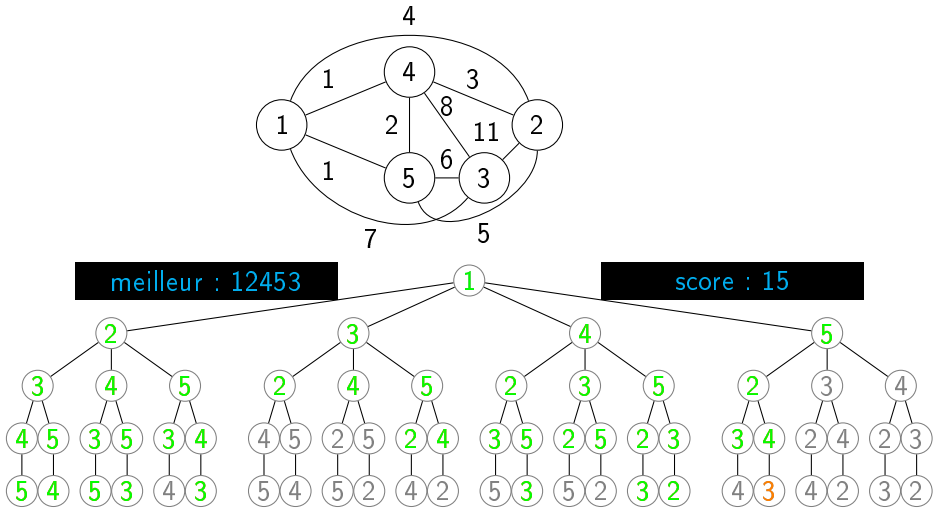
Backtracking



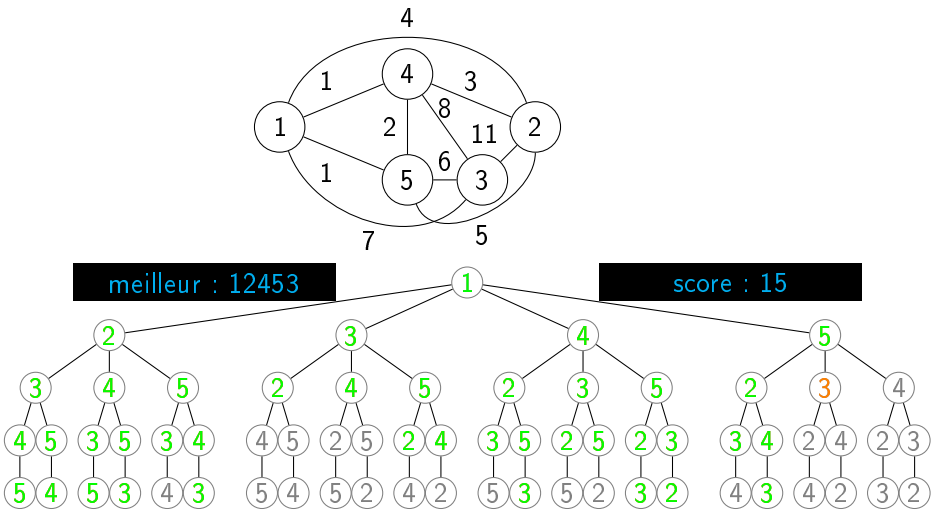
Backtracking



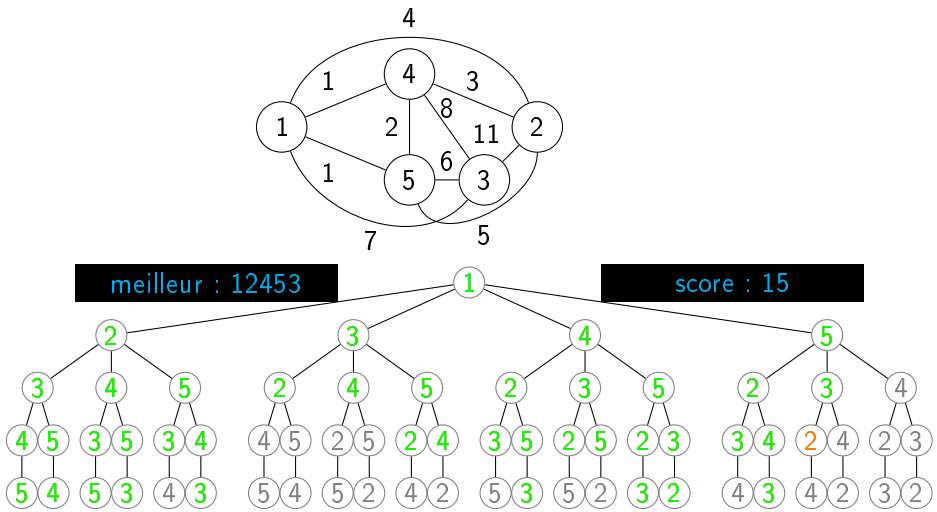
Backtracking



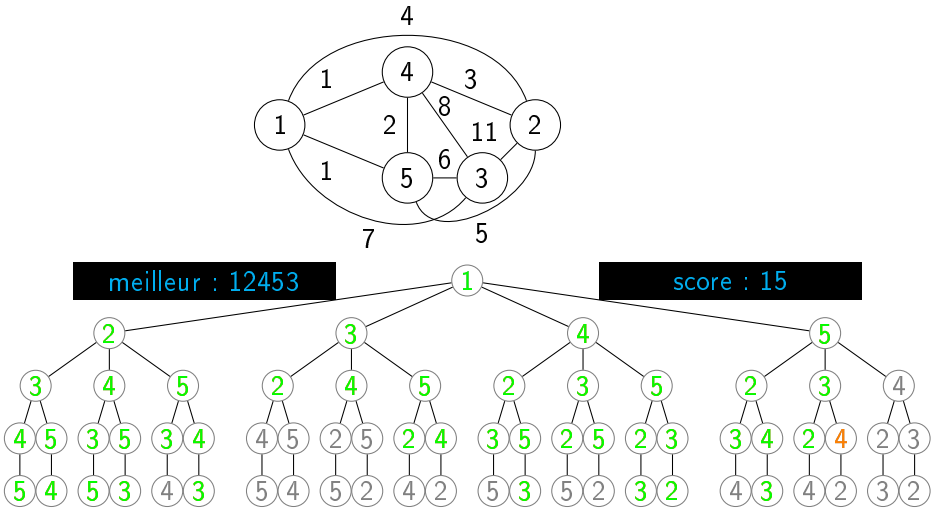
Backtracking



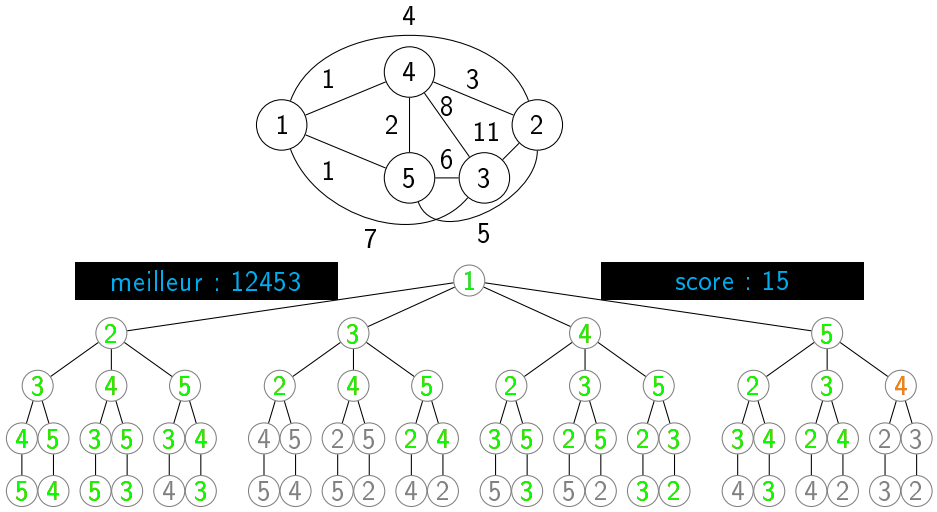
Backtracking



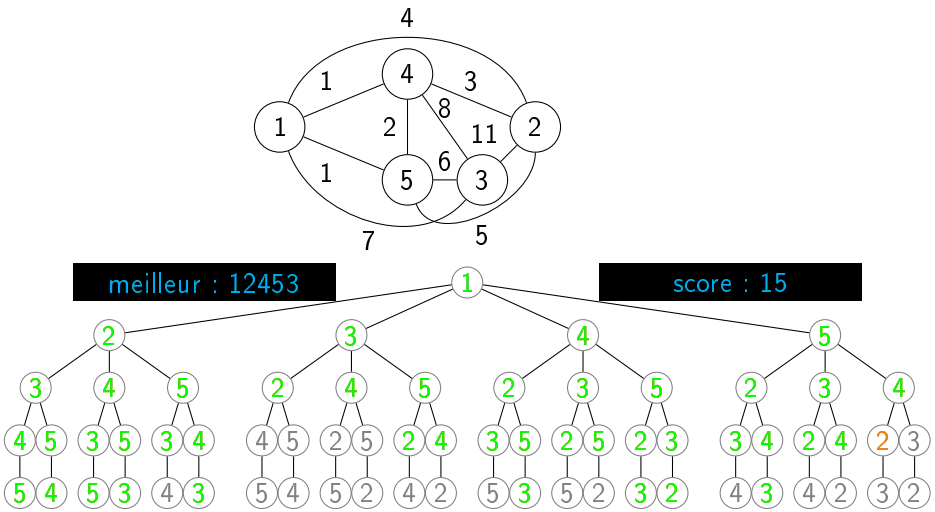
Backtracking



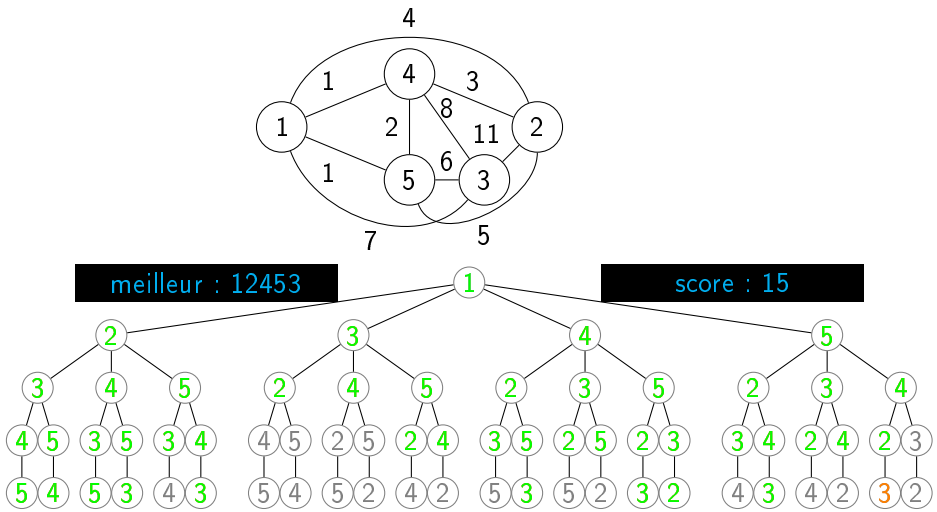
Backtracking



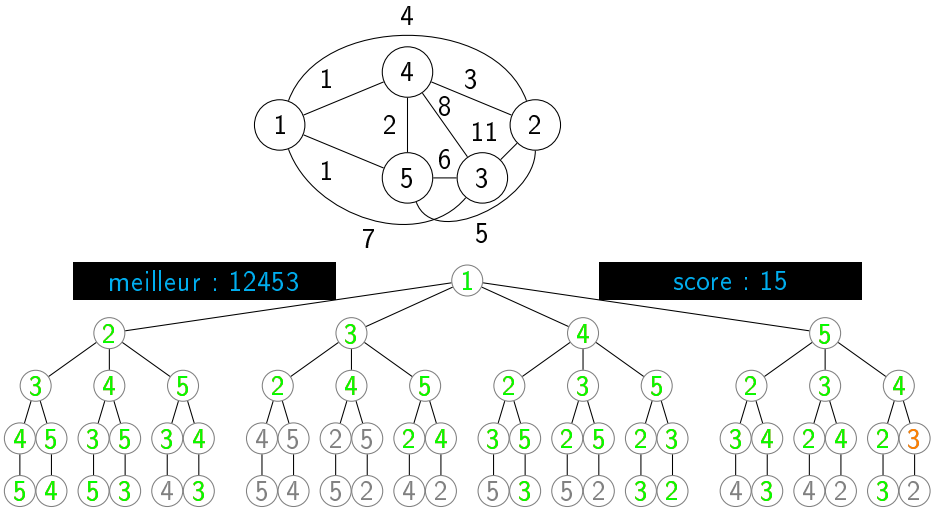
Backtracking



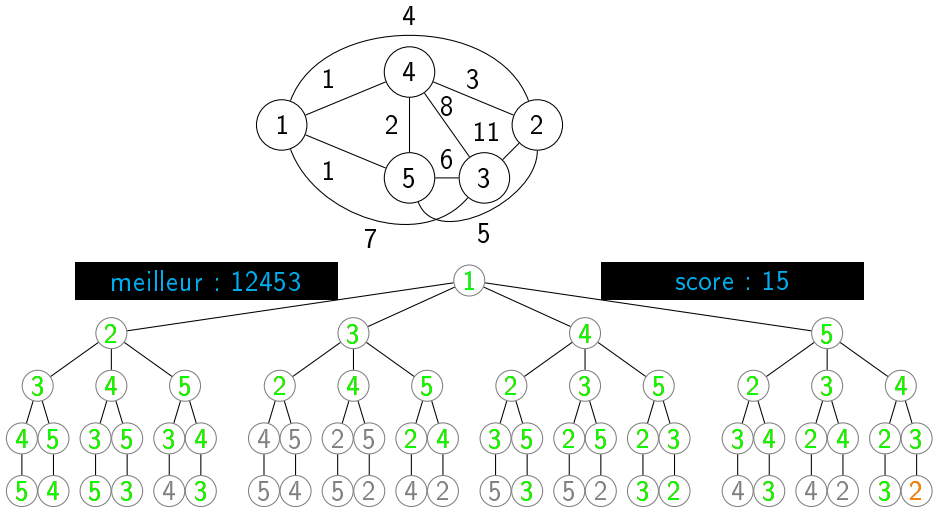
Backtracking



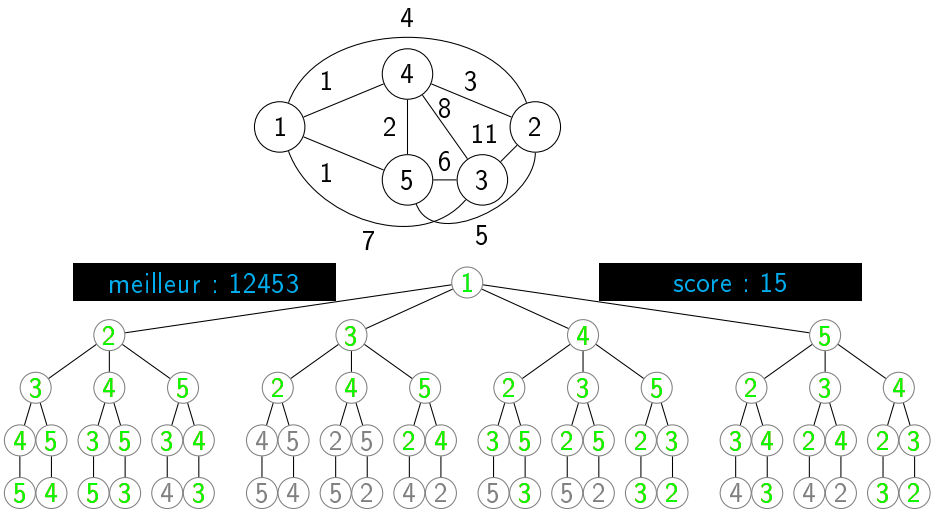
Backtracking



Backtracking

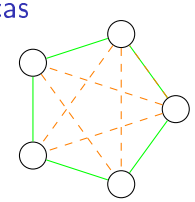


Backtracking



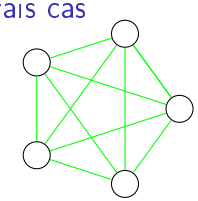
Cas extrêmes

Bon cas



- Peu de chemins intéressants,
- Un choix chanceux des premiers chemins parcourus.

Mauvais cas



- Des valeurs très proches sur toutes les arrêtes,
- Revient alors au parcours exhaustif.

légende

Traits pleins	poids 1
Traits en pointillés	poids 5

Branch and bound

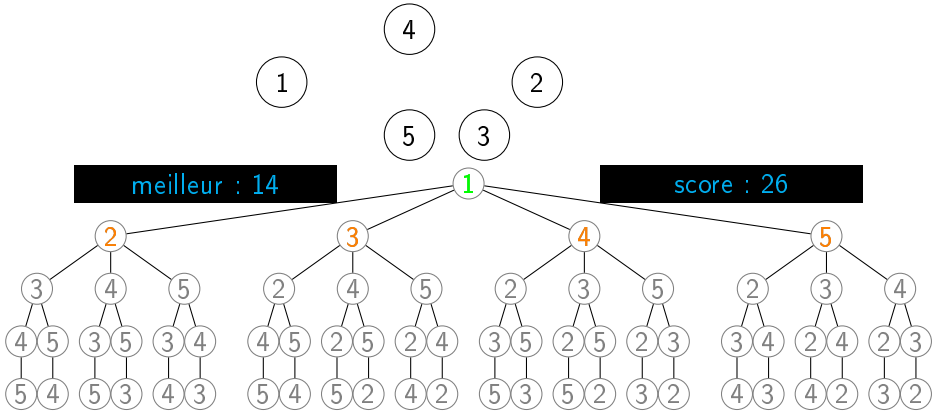
Principe

- Le même principe est appliqué au parcours en largeur,
- Problème : Comment affiner la borne ?
- L'estimation va être affinée à l'aide d'heuristiques...

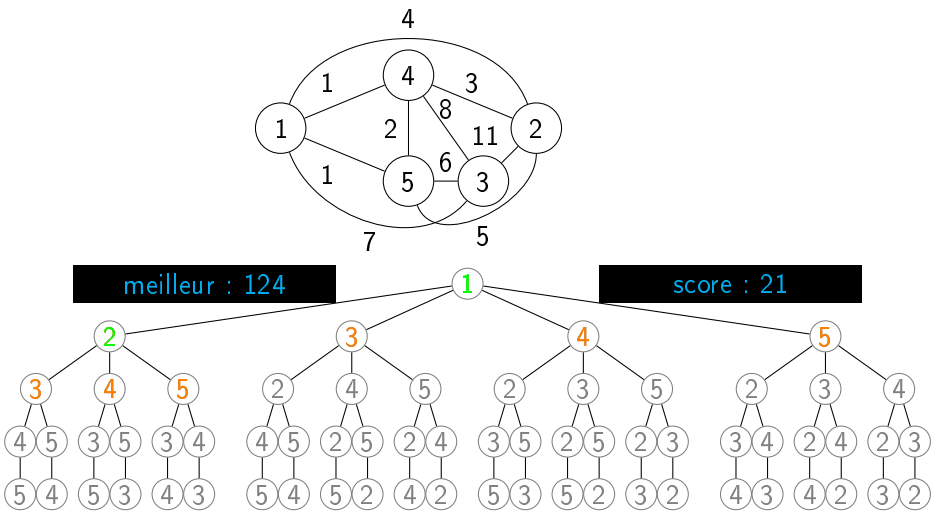
Exemples

- Connaissant la plus petite et la plus grande arrête, il est possible d'encadrer la valeur du plus court chemin et du plus long : coût constant,
- Plus finement, on peut regarder les arrêtes extrêmes dans le sous-graphe restant : coût quadratique,
- Mieux : regarder les arrêtes les plus petites et les plus grandes : coût quadratique.

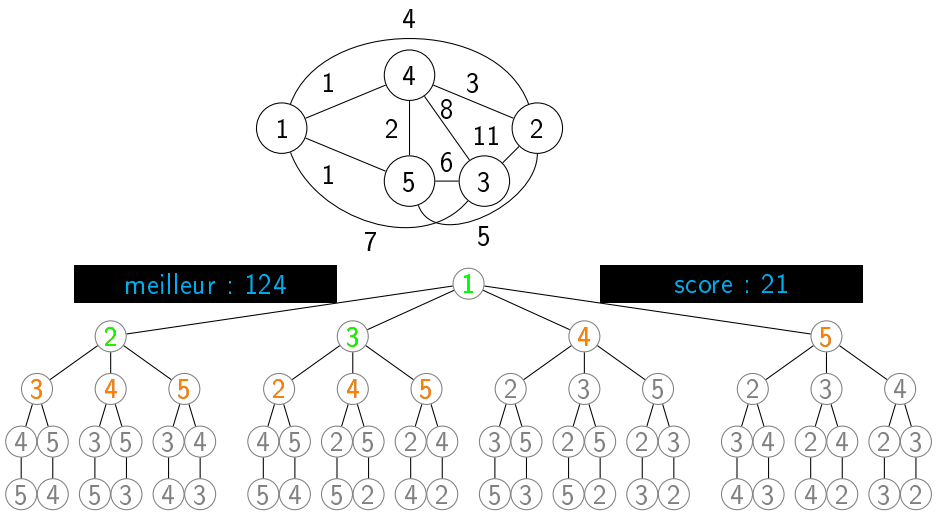
Branch and bound



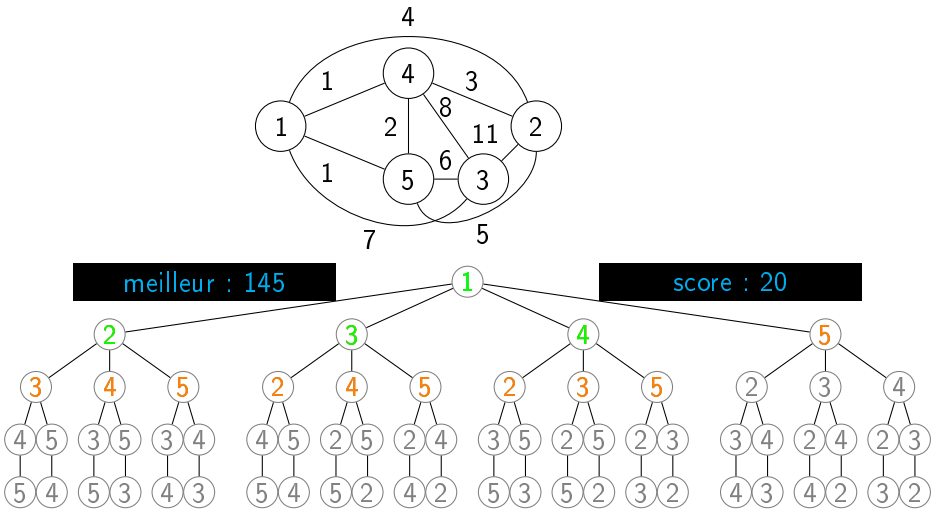
Branch and bound



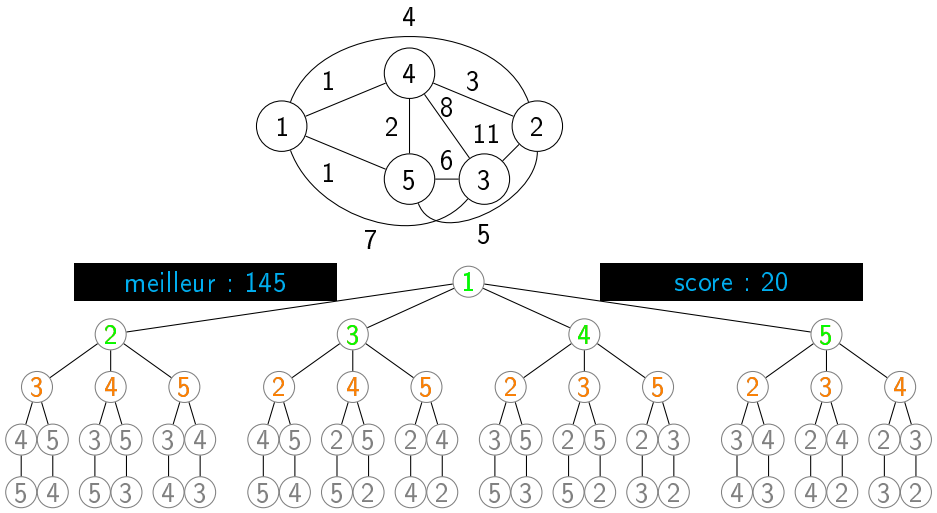
Branch and bound



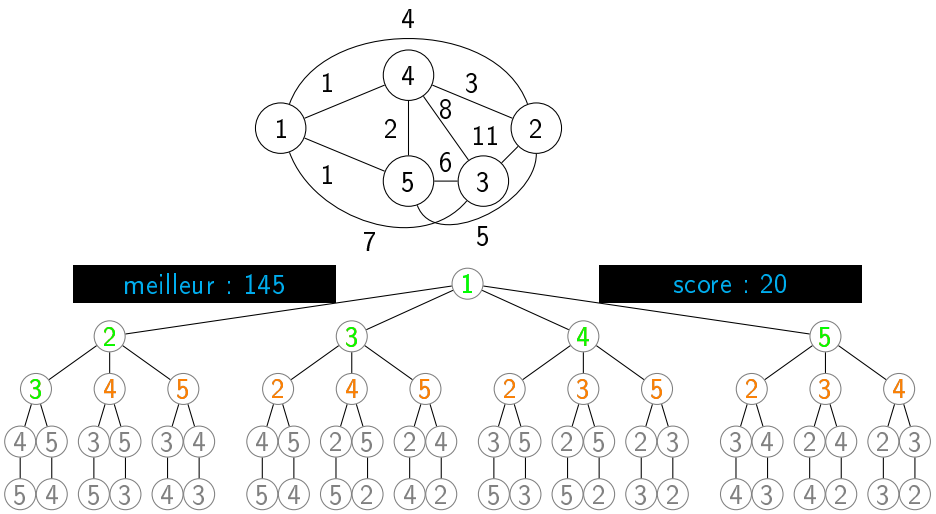
Branch and bound



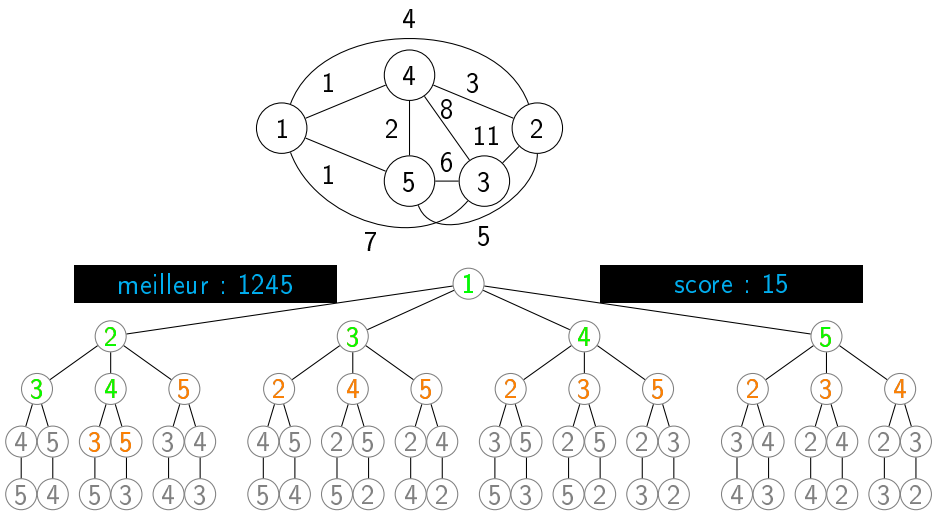
Branch and bound



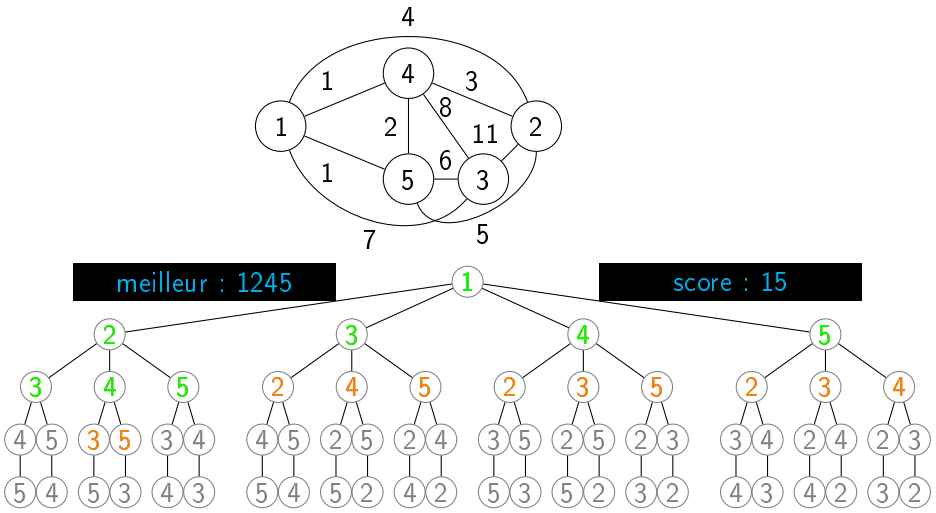
Branch and bound



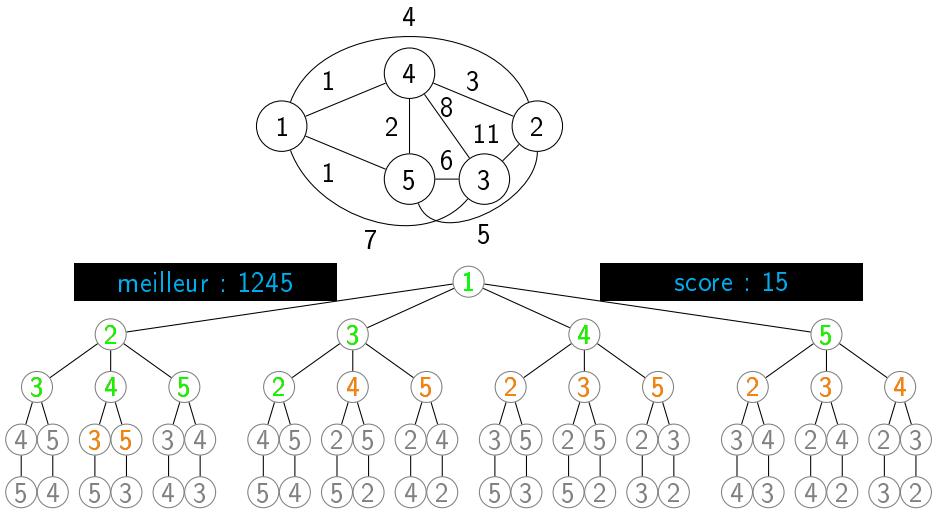
Branch and bound



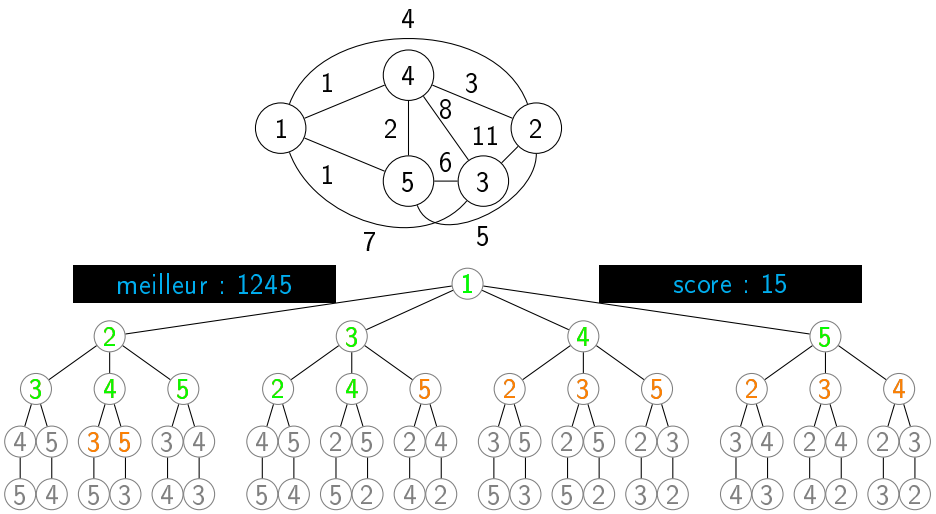
Branch and bound



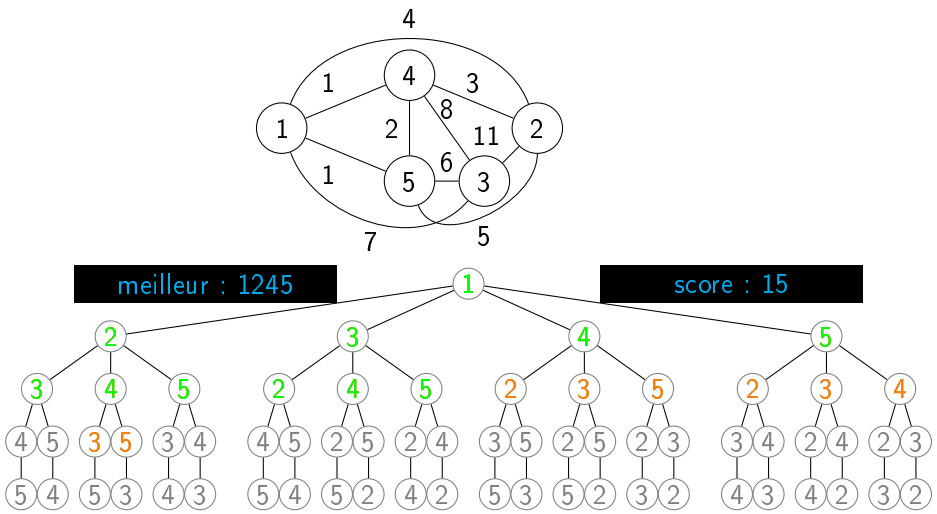
Branch and bound



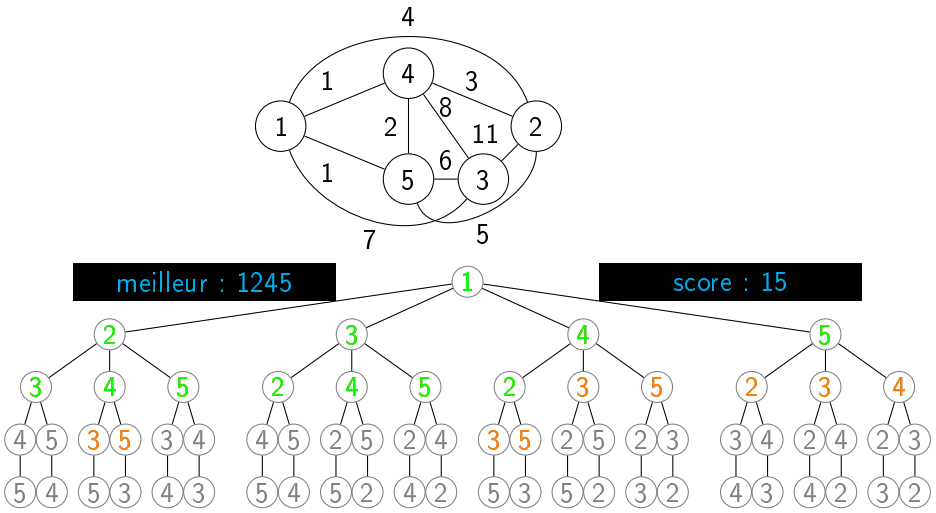
Branch and bound



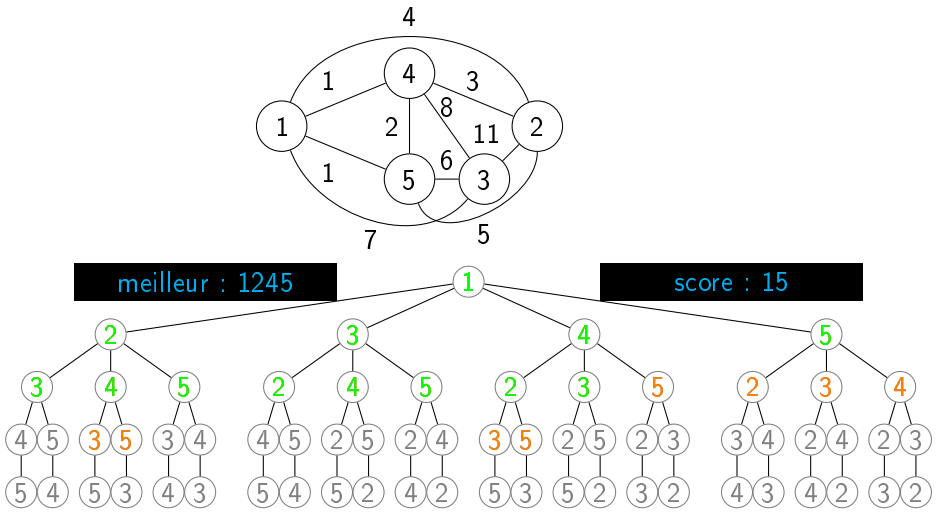
Branch and bound



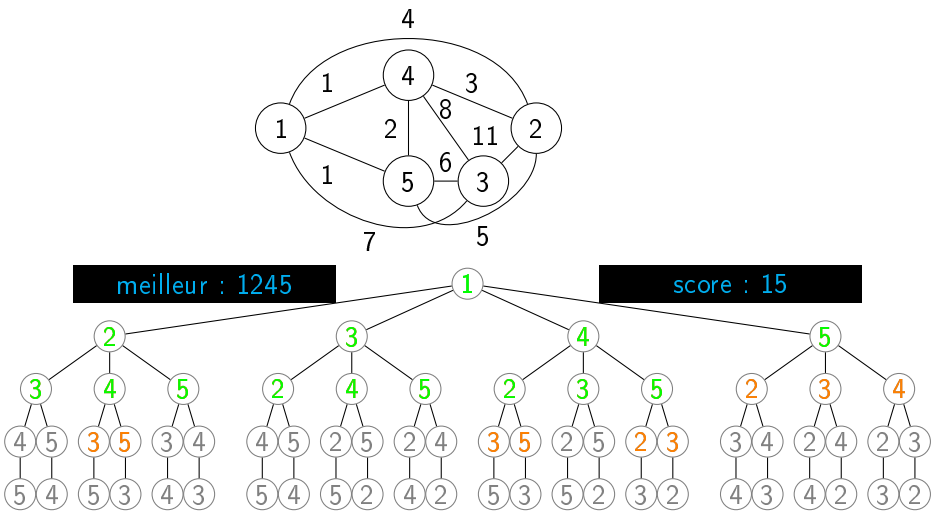
Branch and bound



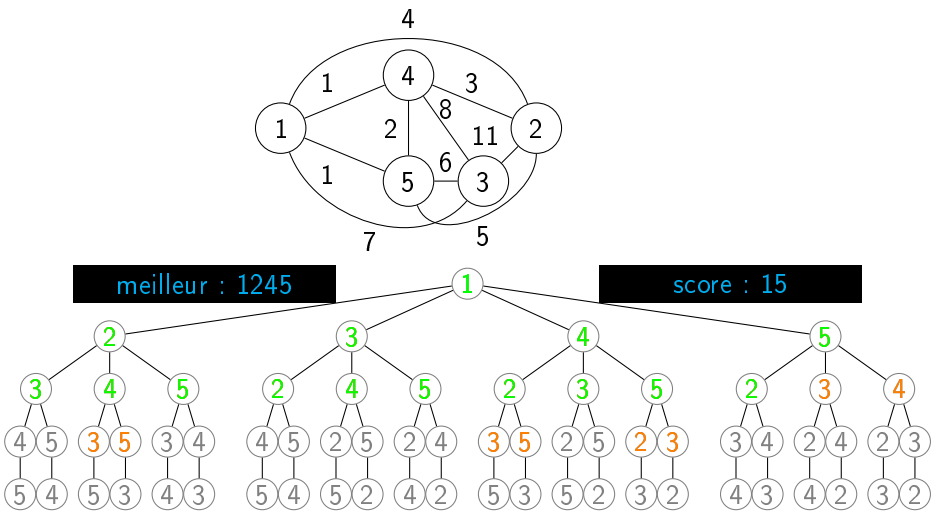
Branch and bound



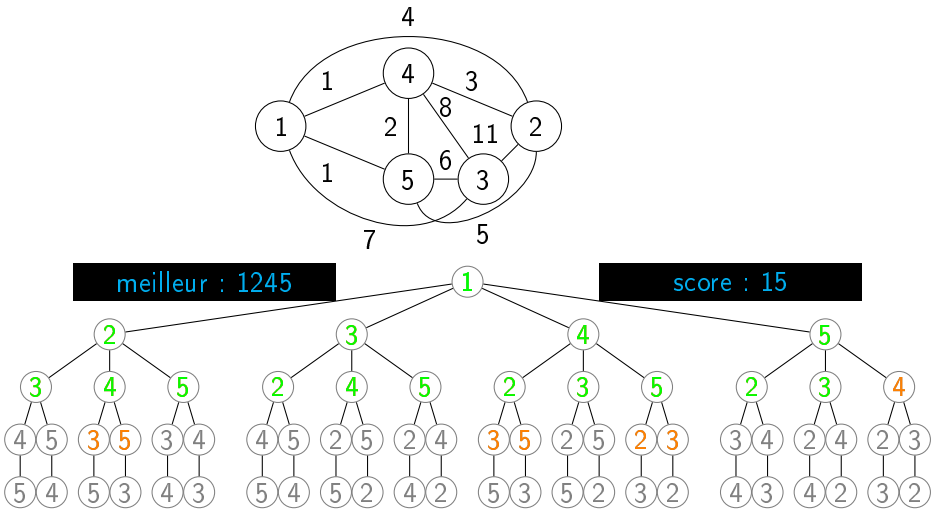
Branch and bound



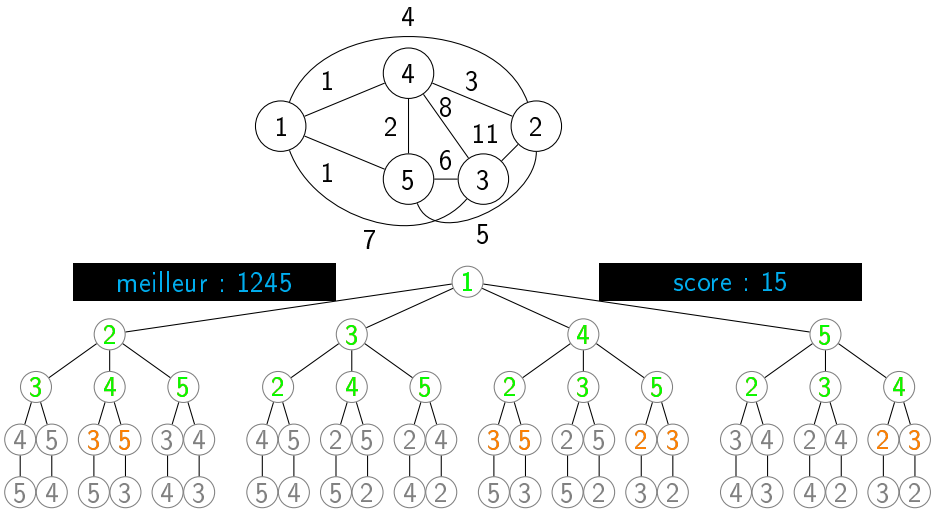
Branch and bound



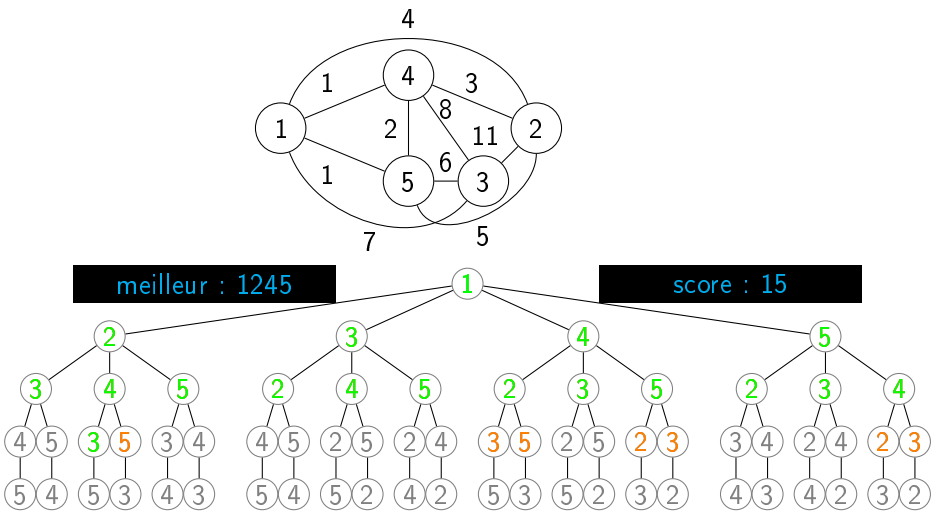
Branch and bound



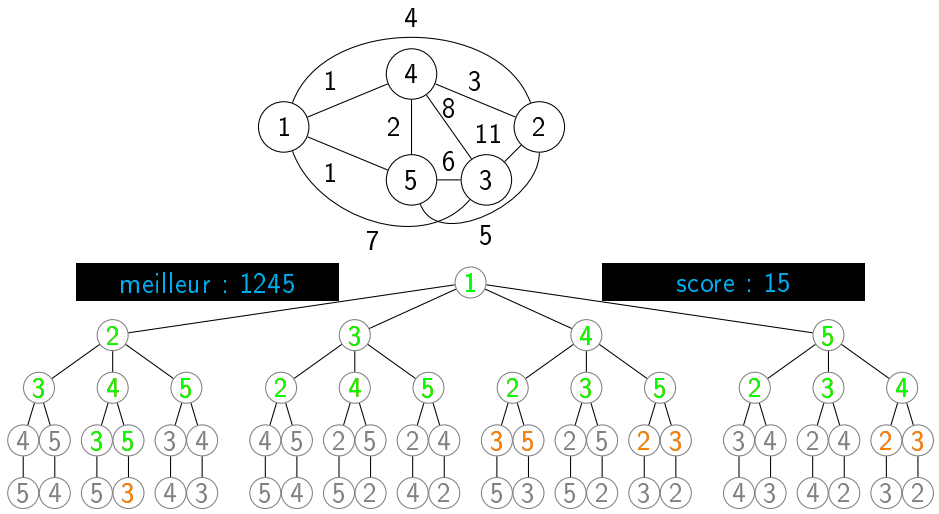
Branch and bound



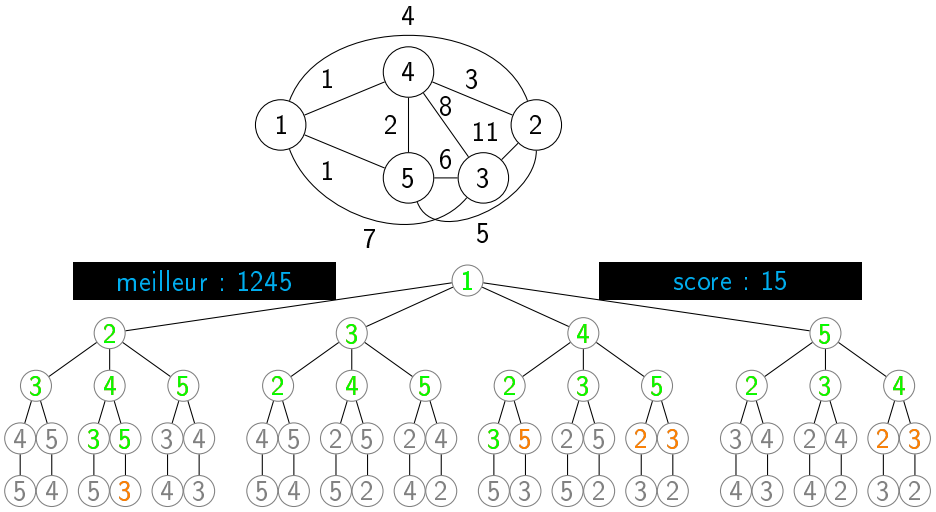
Branch and bound



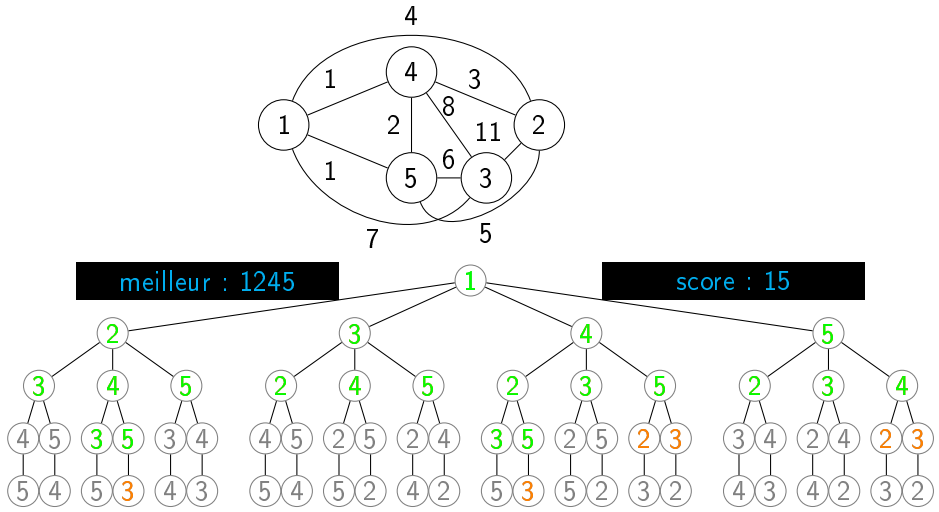
Branch and bound



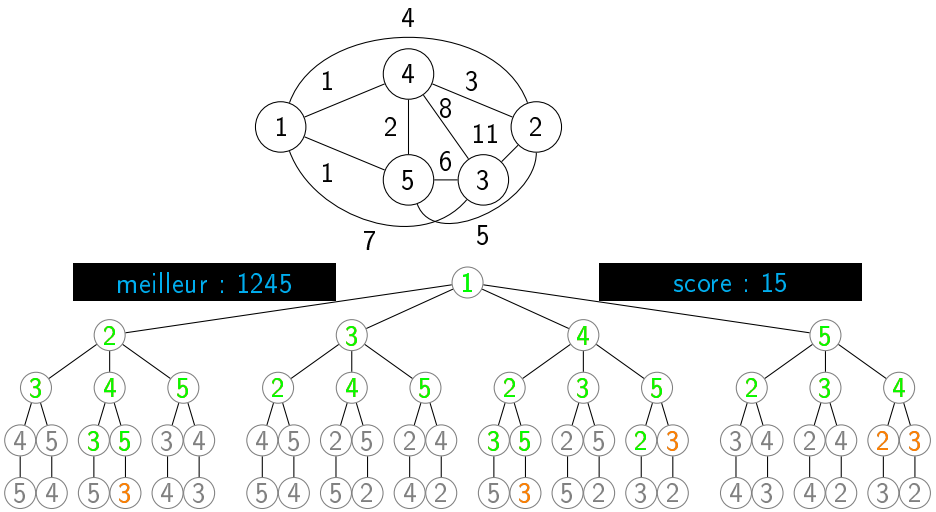
Branch and bound



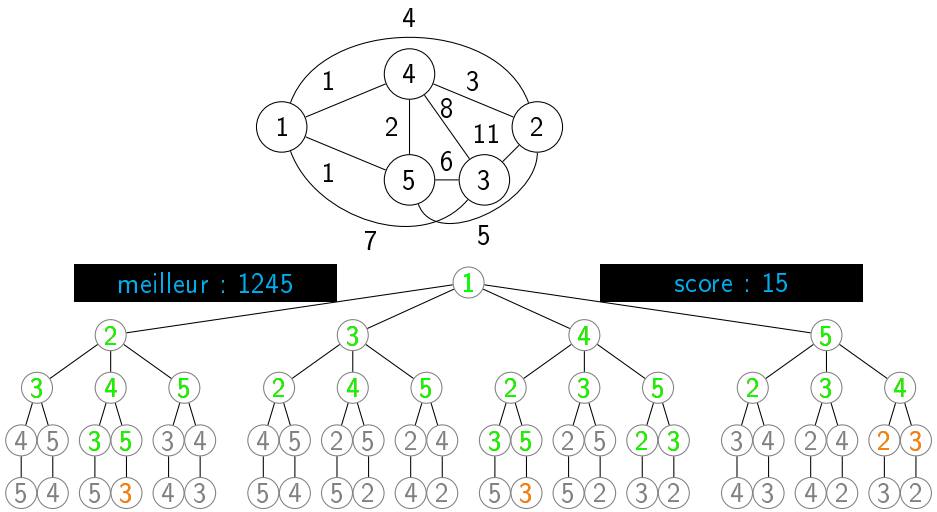
Branch and bound



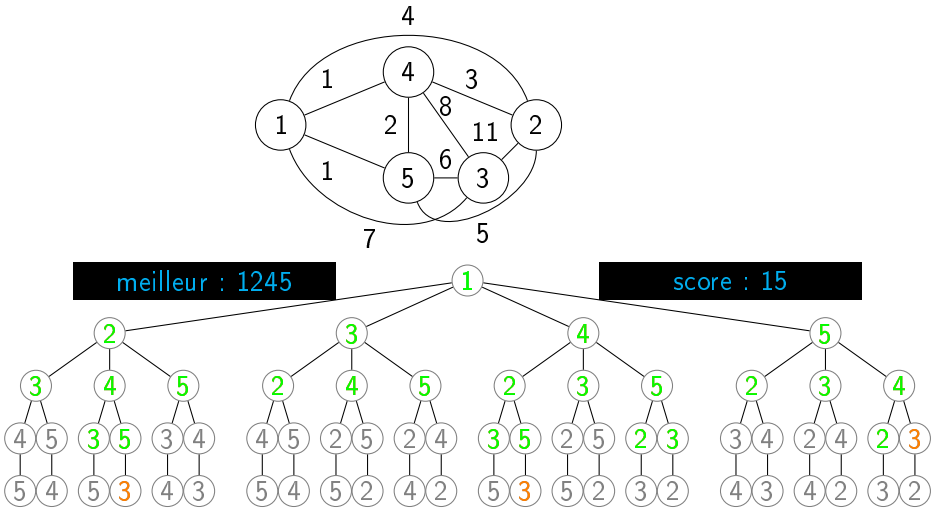
Branch and bound



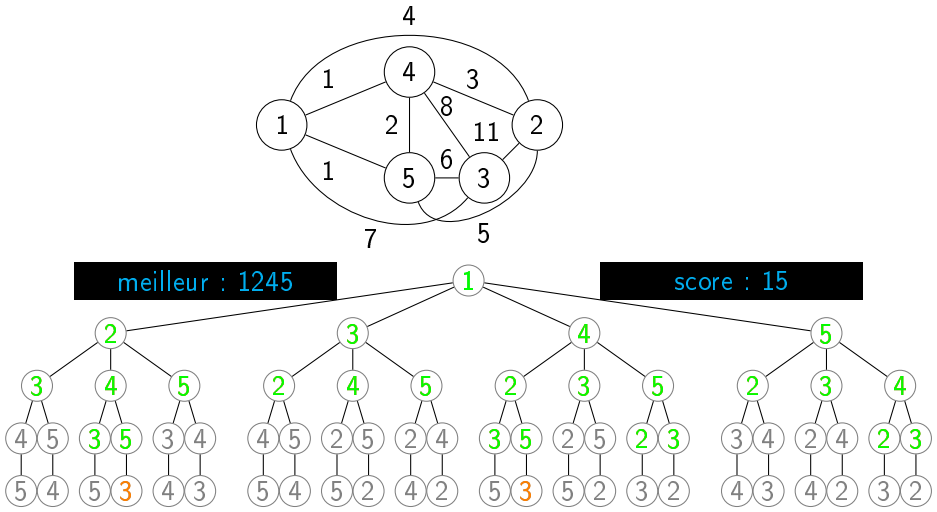
Branch and bound



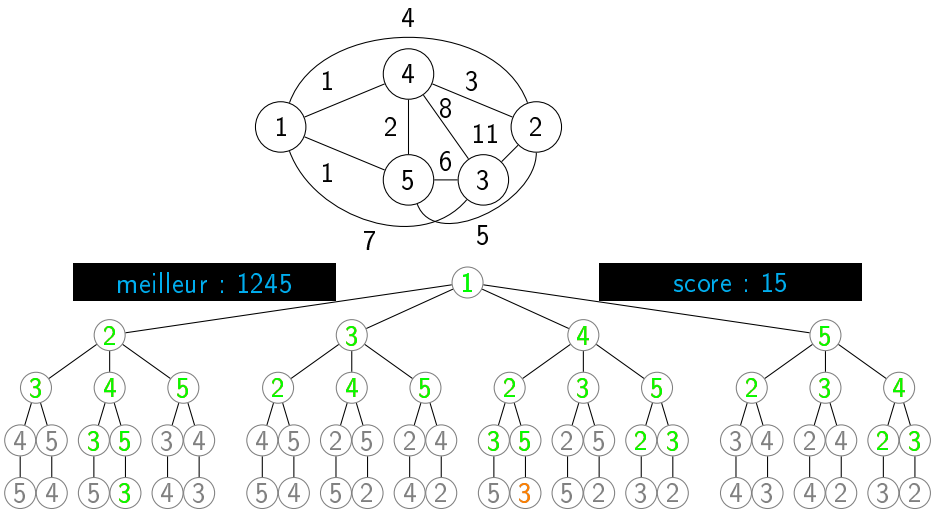
Branch and bound



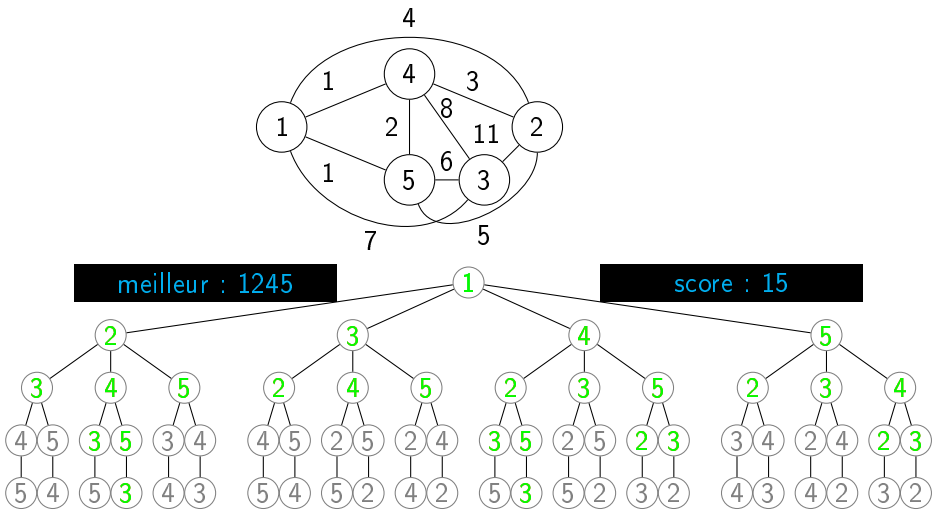
Branch and bound



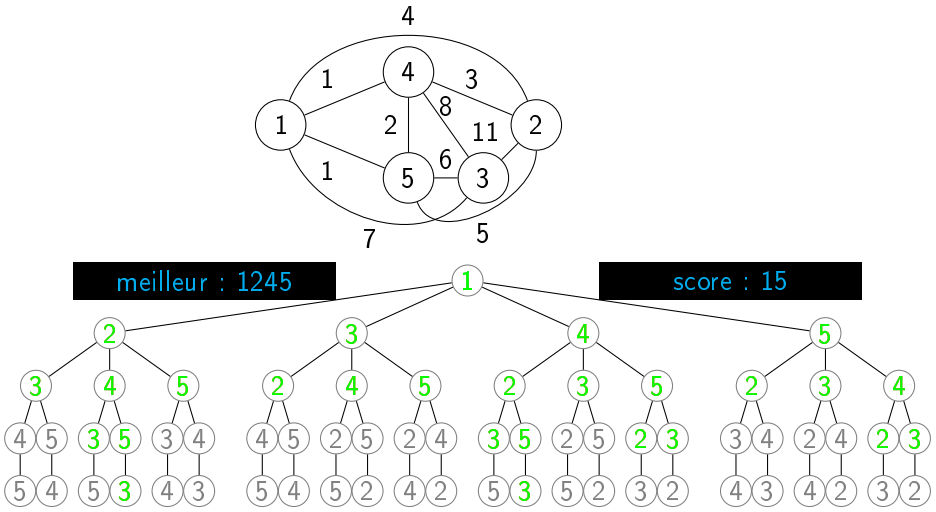
Branch and bound



Branch and bound



Branch and bound



Heuristiques

Différences

- Backtracking : Utile pour le choix des premières branches à visiter, impact indirect sur l'estimation,
- Branch and bound : Permet d'affiner directement l'estimation.

Points communs

- Conditionne l'intérêt de l'algorithme :
  - Trop couteuse : risque d'allourdir le bilan de complexité,
  - Trop lâche : risque de retarder les bons choix,
- Il faut donc essentiellement trouver un bon compromis !