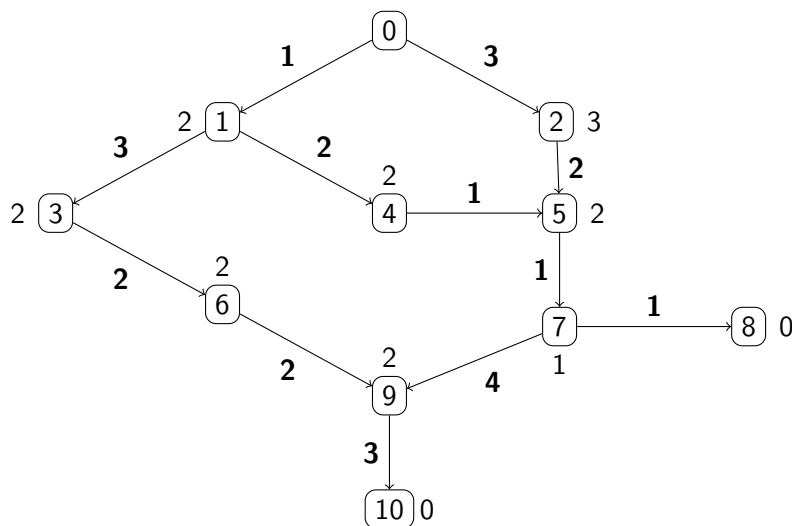


TP 1 - Exploration A^*

Important Ce TP est à déposer sur eCampus avant la prochaine séance de TP.

Implémentation de l'exploration A^*

On considère le graphe de la figure ci-dessous. Les valeurs à côté des sommets indiquent leur valeur de fonction heuristique. Les valeurs en gras à côté des arcs indiquent les coûts des arcs. Les états finaux sont les états 8 et 10.



- Récupérez le fichier `a_star.py` sur eCampus et modifiez/complétez les fonctions `a_star`, `insertion` et `trouver_chemin` pour résoudre l'instance ci-dessus par une exploration A^* .
- Donnez les étapes du déroulement de l'algorithme sous forme de tableau

noeud développé	noeuds ouverts (ordonnés)	noeuds fermés

Application au tri par minimums successifs

Dans un second fichier, récupérez le code écrit précédemment et modifiez-le pour résoudre le problème suivant :

On considère une liste $a = [a_1, \dots, a_n]$ dont les éléments sont les n premiers entiers rangés dans un ordre quelconque. On souhaite trier cette liste dans l'ordre croissant en effectuant le moins d'opération possible. Pour cela, on s'autorise uniquement des opérations d'inversions entre 2 éléments (directement) voisins mal ordonnés.

- Proposez et testez 2 heuristiques différentes et comparez le nombre de noeuds développés selon le choix de l'heuristique.
- Tester votre code sur la liste $[5, 8, 4, 9, 3, 2, 7, 1, 6]$.