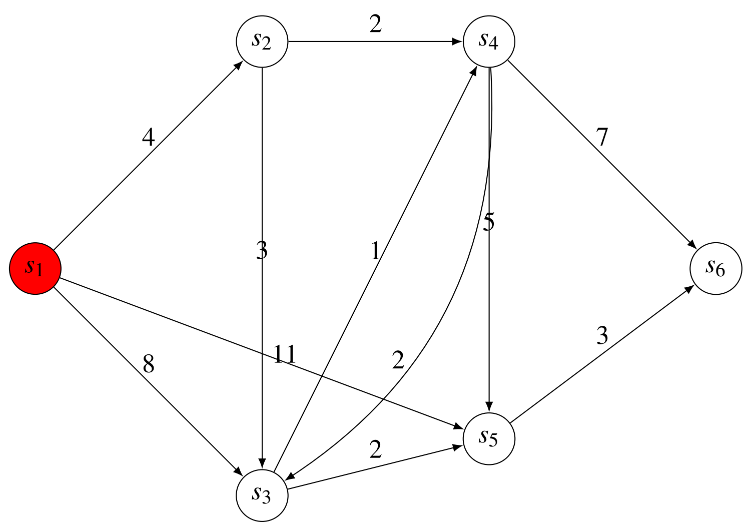
Université de Versailles - Saint Quentin
“Algorithmique de Graphes”
TD7 : Plus courts chemins - Dijkstra

1 Commençons sur un petit graphe

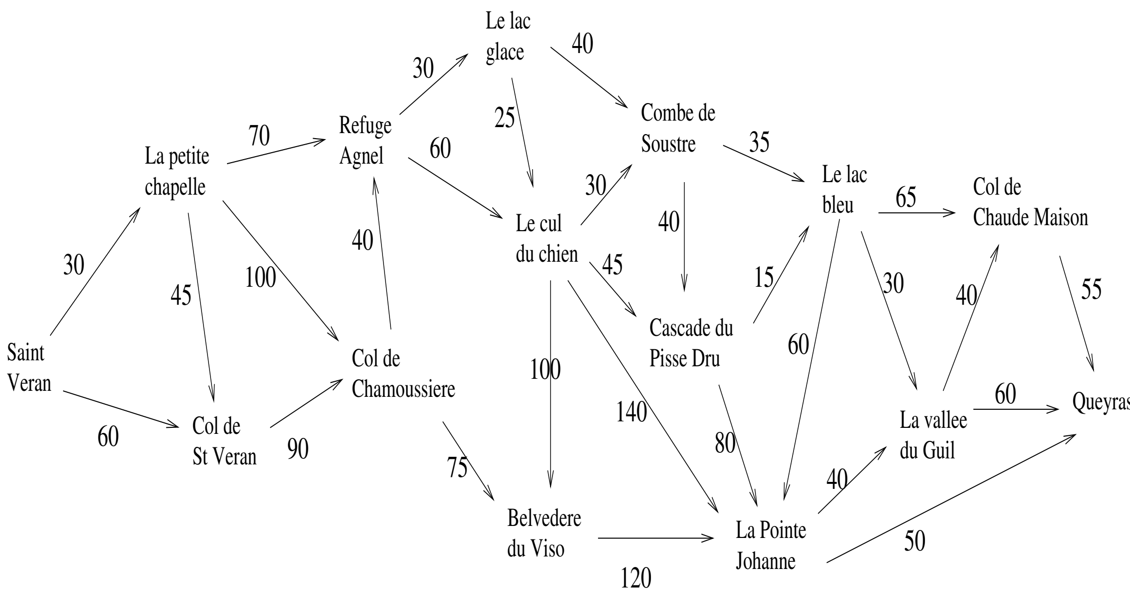
Appliquez l’algorithme de Dijkstra de recherche de plus court chemin à partir du sommet s_1 sur l’exemple ci-dessous. Vous n’indiquerez que :

- le sommet traité à chaque itération.
- l’évolution des structures d et *Pere*
- l’arborescence des plus courts chemins obtenue
- le plus court chemin finalement obtenu pour aller de s_1 en s_6.



2 Randonnées en montagne

En vacances dans les Alpes à Saint Véran, vous décidez de faire une belle randonnée d'une journée en famille pour aller jusqu'à Queyras. Ayant investi dans tous les topo-guides et cartes IGN de la région, vous dessinez le joli graphe que l'on retrouve ci-dessous. Vous y avez en outre mentionné les temps (indiqués en minutes) que vous estimez pour chaque tronçon (remarque : cette carte a été inventée par l'auteur avec des noms plus ou moins réels et ne correspond pas à la réalité - désolé)



1. Vous connaissez vos petits frères (et/ou petites soeurs), vous savez bien que vous allez avoir droit très rapidement à des *"C'est bientôt qu'on arrive ?", "Pfff, c'est long", "On est loin là encore ?" ...* Et vous décidez donc de rechercher l'itinéraire le plus rapide. Quel est-il ?
2. Changement de décor : vous êtes toujours en famille, mais ce coup-ci avec les parents de votre copine (ou copain) qui font les malins et se présentent comme des randonneurs infatigables. Vous décidez donc de rechercher l'itinéraire le plus long. Pouvez-vous adapter l'algorithme que vous avez utilisé à la première question ? Et si non, auriez-vous une idée de comment faire ?

3 Le chemin le plus court pour aller d'un point à un autre, c'est encore de ne pas y aller (exercice partiel 2019)

Soit un graphe $G = (V,A)$ orienté et pondéré avec des poids strictement positifs $w(u,v)$ sur chaque arc (u,v) et un noeud source s. L'algorithme de Dijkstra calcule une valeur D_i pour chaque noeud i correspondant à la valeur du plus court chemin de s à i. On souhaite ajouter à cet algorithme de Dijkstra (rappelé ci-dessous) le compte du nombre de plus courts chemins de s vers chaque noeud i. Il s'agit donc de compléter l'algorithme de Dijkstra pour y ajouter le calcul de N_i qui doit contenir le nombre de plus courts chemins de s à i.

- Initialisations :
  - $T = \{s\}$; $D_s = 0$;
  - $\forall i \neq s$, si l'arc (s,i) existe alors $D_i = w(s,i)$ (∞ sinon)
- Boucle principale : Tant que $T \neq V$ faire
  - Trouver un noeud t de $V - T$ tq $D_t = Min(D_i, i \in V - T)$,
  - $T = T \cup \{t\}$
  - $\forall k \in \Gamma_t^+$, si $(D_k > D_t + w(t,k))$ alors $D_k = D_t + w(t,k)$

1. Réécrire l'algorithme de Dijkstra en y insérant le calcul de ces valeurs N_i.
2. L'appliquer sur le graphe ci-dessous. Vous détaillerez bien les différentes itérations et comment D et N évoluent.
3. Finalement, quels sont les différents plus courts chemins pour aller de s à H ?