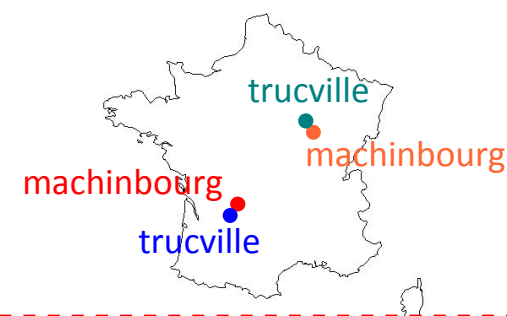


Exercice : isotoponymes et villes confondues



Tester d'abord sur un petit exemple artificiel

0) Récupérer le fichier **villes2.zip** avec les noms et coordonnées de 35180 villes de France métropolitaine. Les charger dans un **vector<Town>** (code fourni).

Il se trouve que certaines villes ont des **noms identiques**. De plus, du fait d'approximations, certaines villes ont aussi des **coordonnées identiques**.

map et set
suffisants, mais
unordered_XXX
aussi possibles

- 1) Utiliser une table associative pour compter, construire et afficher sous forme de texte l'**histogramme des répétitions de noms de villes** [= le nb de noms de villes utilisés par 1 ville exactement, 2 villes exactement, 3 villes exactement, ...]. [≈ 15 LOC]
- 2) Afficher l'**histogramme du nb de villes de mêmes coordonnées (Point2D)**. [≈ 15 LOC]
- 3) Calculer l'ensemble N des villes qui ont **une autre ville de même nom** et l'ensemble C des villes qui ont **une autre ville de mêmes coordonnées** [Étendre **Town** pour pouvoir définir **set<Town>**]. Calculer $N \cap C$ avec une complexité $O(|N| + |C|)$ [avec fonction de la STL dédiée à ça]. Combien de villes ont cette propriété conjointe [= $|N \cap C|$] ? [≈ 20 LOC]
- 4) Calculer efficacement pour combien de villes on peut se tromper en entendant parler d'**une ville A toute proche d'une ville B** ? [= nb de villes $v1$ tq $\exists v2, v3, v4$ tq $\text{coord}(v1) = \text{coord}(v2)$, $\text{nom}(v1) = \text{nom}(v3)$, $\text{coord}(v3) = \text{coord}(v4)$, $\text{nom}(v2) = \text{nom}(v4)$] Comparer à l'approche naïve (35180^4 cas $\approx 10^{18}$, mais que les premiers tests d'égalité élaguent beaucoup toutefois : à combien ?) : quel est le gain de temps ? [$> \times 300$]

Ne pas utiliser find ni sort. Ne pas modifier les structures de données sauf si c'est indispensable pour les opérations présentées en cours