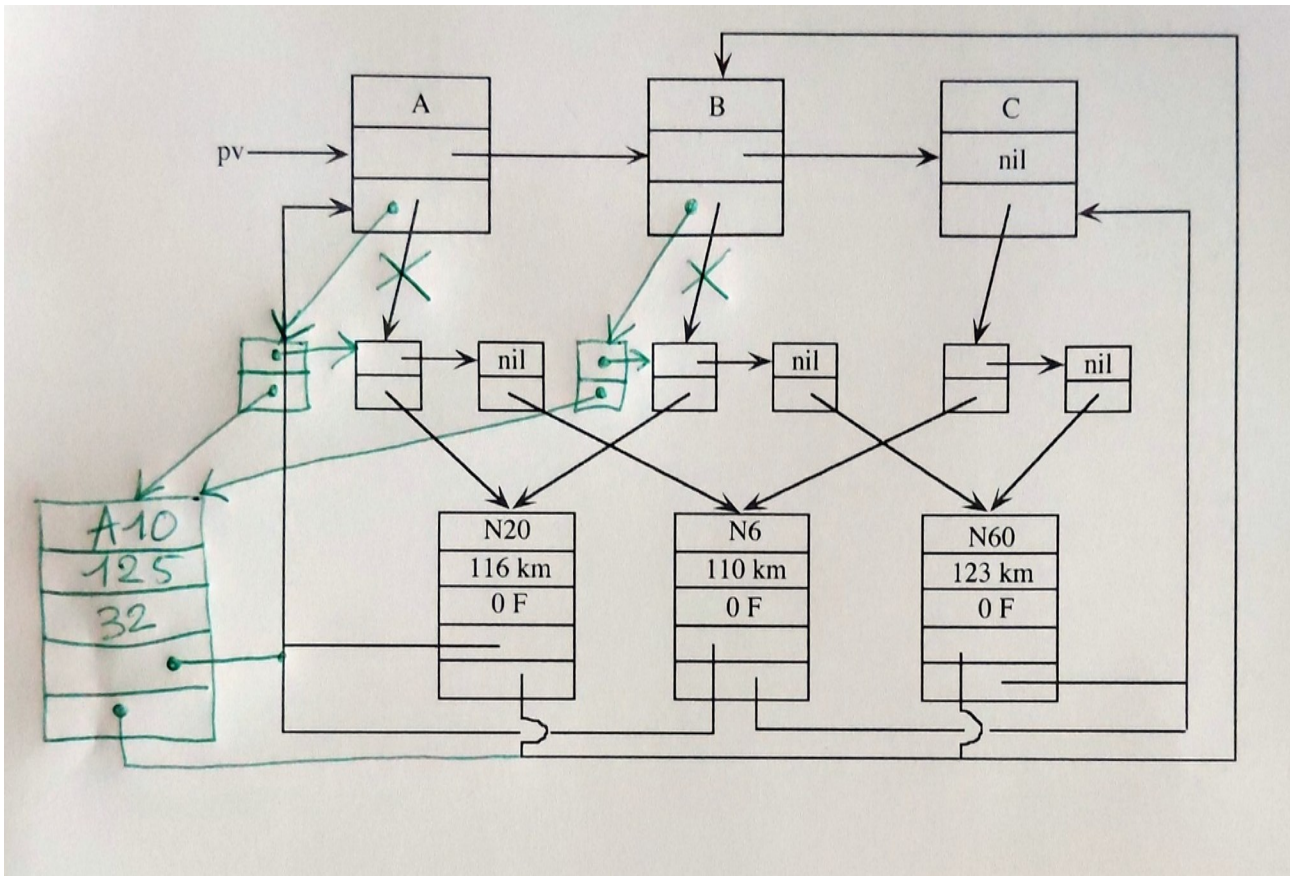
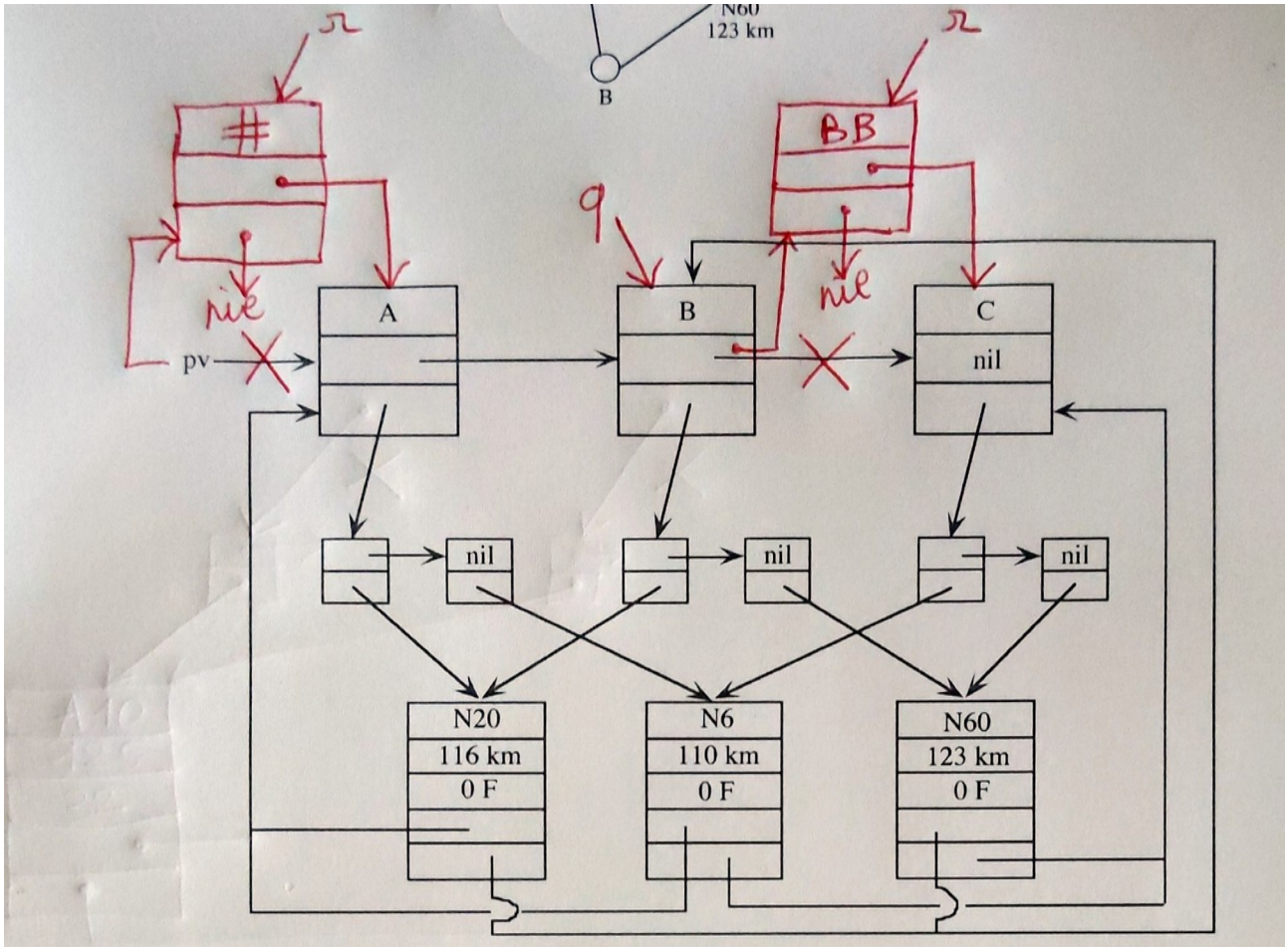
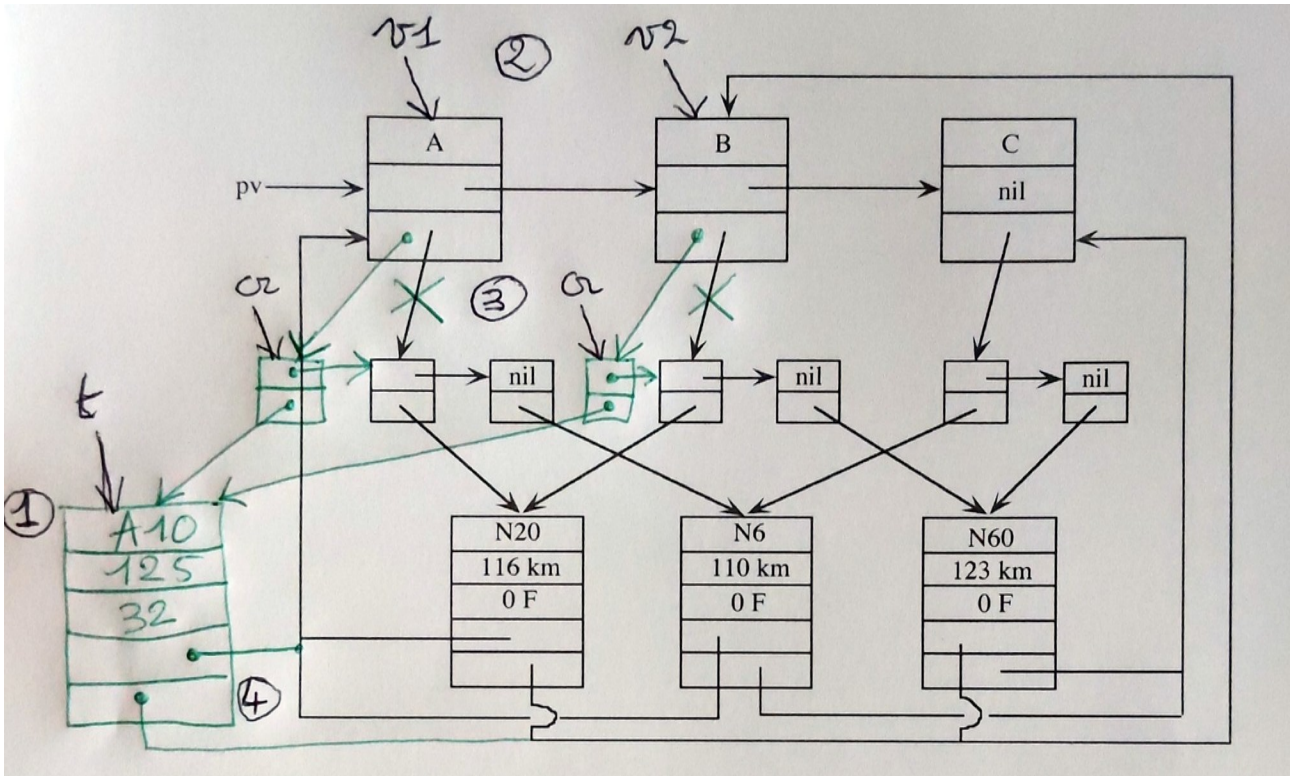
TD10 - Liste villes

```
Type   listeVille = ^ville
       listeConRou = ^conRou
       listeTron = ^tron
       ville = Enregistrement
                    nom : chaine
                    suiv : listeVille
                    connex : listeConRou
               FinEnregistrement
       conRou = Enregistrement
                    suiv : listeConRou
                    tron : listeTron
                FinEnregistrement
       tron = Enregistrement
                    route : chaine
                    dist, peage: entier
                    orig, extrem : listeVille
                FinEnregistrement
```






```
Procédure ajouteVille (E v : Chaine, E/S pv : listeVille, S r : listeVille)
Var q : listeVille
Début
  r ← allouer(ville)
  r^.nom ← v
  r^.connex ← nil
  Si (pv=nil) ou (v<pv^.nom)
       Alors r^.suiv ← pv
             pv ← r
       Sinon
       q ← pv
       TantQue (q^.suiv≠nil) et (q^.suiv^.nom<v) Faire
             q ← q^.suiv
       FinTantQue
       r^.suiv ← q^.suiv
       q^.suiv ← r
  FinSi
Fin
```




```
Procédure ajouteLiaison (E o,e,r : Chaine, d,p : entier, E/S pv : listeVille)
Var t : ^tron
    v1, v2, q : ^ville
    cr : ^conRou
    villemax : chaine
Début
  (*  1  *)
  t ← allouer(tron)
  t^.route ← r
  t^.dist ← d
  t^.peage ← p
  (*  2  *)
  v1 ← nil
  v2 ← nil
  q ← pv
  Si o<e alors villemax ← e
        sinon villemax ← o
  FinSi
  TantQue (q≠nil) et (villemax >= q^.nom) Faire
       Si q^.nom = o
          Alors v1 ← q
          Sinon Si q^.nom = e
                     Alors v2 ← q
                FinSi
       FinSi
       q ← q^.suiv
  FinTantQue
  Si v1 = nil
       Alors ajouteVille(o,pv,v1)
  FinSi
  Si v2 = nil
       Alors ajouteVille(e,pv,v2)
  FinSi
  (*  3  *)
  cr ← allouer(conRou)
  cr^.tron ← t
  cr^.suiv ← v1^.connex
  v1^.connex ← cr
  cr ← allouer(conRou)
  cr^.tron ← t
  cr^.suiv ← v2^.connex
  v2^.connex ← cr
  (*  4  *)
  t^.orig ← v1
  t^.extrem ← v2
Fin
```