**Document réponses Numérique et Sciences Informatiques**

|  |
| --- |
| **I-1.** Ordre 1 : (r1), (r2), (r3), (r4), (r5)  pas d’erreur   erreur  *à utiliser seulement si vous vous êtes trompé au-dessus* *pas d’erreur* *erreur*  Ordre 2 : (r1), (r5), (r4), (r3), (r2)  pas d’erreur   erreur  *à utiliser seulement si vous vous êtes trompé au-dessus* *pas d’erreur* *erreur*  Ordre 3 : (r2), (r3), (r4), (r5), (r1)  pas d’erreur   erreur  *à utiliser seulement si vous vous êtes trompé au-dessus* *pas d’erreur* *erreur*  Ordre 4 : (r5), (r1), (r4), (r2), (r3)  pas d’erreur   erreur  *à utiliser seulement si vous vous êtes trompé au-dessus* *pas d’erreur* *erreur* |
| **I-2.** (a)        (b) |
| **I-3.** …… …… ……  …… |
| **I-4.** …… ……  ……  …… |
| **I-5.** ……  …… …… …… |
| **II-1.** …… …… ……  …… …… |
| **II-2.** …… ……  …… ……  …… …… |
| **II-3.** def meilleur\_chemin(base,depart,arrivee):  arcs = creer\_dico\_arcs\_sortants(base)  a\_visiter = {p: (inf, '', '') for p in arcs.keys()}  a\_visiter[depart] =  visites =  while a\_visiter != :  # recherche du sommet suivant à visiter  p =  (dist,precedent,sentier) = a\_visiter[p]  # m-à-j des voisins de p restant à visiter  for (suivant,n,d) in :  if in a\_visiter :  (min,prec,sent) =  poids = + dist  if < :  a\_visiter[suivant]=( ,  ,  )  # p passe des sommets à visiter aux sommets visités  visites[p] = a\_visiter[p]  del a\_visiter[p]  affichage(visites, depart, arrivee) |
| **II-4.**  ligne  ligne   ligne  ligne |