
TP n°3 - Autour du tri rapide

Notions abordées

- Tri rapide en place
- Invariant de boucles
- Génération aléatoire de liste d'entiers, de caractères
- Produire d'une part une librairie de fonctions de tri, et d'autre part des tests pour ces fonctions.

 Toutes les fonctions doivent être commentées et testées. Elles doivent notamment être munies d'une description faisant suite à d'éventuelles hypothèses sur leurs arguments placée entre triple guillemets sous la signature de la fonction, (`""" documentation sur plusieurs lignes """`). Les autres commentaires s'écrivent sur des lignes commençant par dièse : `# ligne de commentaire`.

Exercice 1 Tri rapide

Question 1

Après un exemple fait au tableau, écrire en pseudo-code l'algorithme de tri rapide. Préciser les invariants de boucles nécessaires à la preuve de correction de la fonction.

Question 2

Définir une fonction récursive `aux_tri_rapide` qui prend en argument une liste, et deux indices `deb` et `fin`, et qui trie la liste entre les indices `deb` et `fin-1`. *On veillera à bien expliquer le rôle des variables locales, à travers l'explicitation d'un invariant de boucle. De plus on veillera à ne pas modifier les arguments `deb` et `fin`.*

Question 3

Définir une fonction récursive `tri_rapide_en_place` qui prend en argument une liste, et la trie par tri rapide.

Question 4

Recopier la fonction `est_croissante` proposée en classe (et disponible sur cahier de prépa) afin de tester la fonction de tri ci-dessus sur quelques exemples générés à la main.

Dans l'exercice suivant on s'intéresse à la génération aléatoire de liste afin de pouvoir tester nos fonctions sur des exemples plus grands, et plus représentatifs.

Exercice 2 Générer des listes aléatoires

Le module `random` propose une fonction `randint` qui prend en argument deux entiers définissant un intervalle non vide, et qui renvoie un entier tiré aléatoirement dans cet intervalle.

Question 1

Faire plusieurs appels à `randint` dans l'interpréteur et constater son comportement aléatoire.

Question 2

Définir une fonction `liste_random` qui prend en argument trois entiers `n`, `vmin` et `vmax`, et qui calcule une liste de `n` entiers choisis aléatoirement dans l'intervalle `[vmin..vmax]`. *Vous veillerez à préciser les hypothèses que ces trois entiers doivent vérifier pour que cette description ait un sens.*

Question 3

Comme l'évaluation d'un appel à `liste_random` est aléatoire, on ne peut tester la fonction par des tests d'égalité avec un résultat attendu. En revanche le résultat attendu n'est pas quelconque, la liste produite doit vérifier certaines propriétés. Lister lesquelles et en déduire un jeu de tests de `liste_random`.

Question 4

Définir `liste_random_comp` une fonction analogue à `liste_random` dans laquelle la liste est créée par compréhension.

Question 5

Définir une fonction `liste_taille_random` qui prend en argument trois entiers `nmax`, `vmin` et `vmax`, et qui calcule une liste de `n` entiers choisis aléatoirement dans l'intervalle `[vmin..vmax]`, où `n` est lui même choisi aléatoirement entre 0 et `nmax`. *Vous veillerez à préciser les hypothèses que ces trois entiers doivent vérifier pour que cette description ait un sens.*

Question 6

En utilisant la fonction `chr` proposer une fonction qui crée une liste aléatoire de caractères. *On vous laisse décider des paramètres pertinents pour cette fonction, et expliciter les hypothèses*