

TP4 : *Jeu de bridge*

À rendre avant 12h le 18 novembre 2009 par email à votre chargé de TP ¹

Ce TP traite du bridge qui se joue avec un jeu de 52 cartes. Nous n'aborderons que quelques notions de ce jeu compliqué pour pouvoir définir des opérations de tri et d'évaluation.

1 Questions préliminaires

1. Écrire une fonction `min` : `'a list -> 'a * 'a list` qui renvoie le minimum d'une liste selon l'ordre `<=` et le reste de la liste. L'appel `min []` terminera sur une erreur.
2. Écrire une fonction `tri_liste_inf` : `'a list -> 'a list` qui trie par ordre croissant une liste en utilisant la fonction `min`.
3. Écrire une fonction `min_gen` : `('a -> 'a -> bool) -> 'a list -> 'a` qui renvoie le minimum d'une liste en utilisant une fonction de comparaison propre de signature `'a -> 'a -> bool`.
4. Écrire une fonction `tri_liste_gen` : `('a -> 'a -> bool) -> 'a list -> 'a list` qui trie une liste en utilisant `min_gen`.

2 Définition des cartes

Chaque carte a une couleur parmi les 4 suivantes : Pique, Cœur, Carreau, et Trèfle et une valeur pouvant être soit un honneur (as, roi, dame ou valet) soit un entier compris entre 2 et 10.

5. Définir les types `couleur`, `valeur` et `carte`.

Chaque joueur possède 13 cartes. Cet ensemble s'appelle une *main* et sera représenté par une liste de cartes. Les cartes au bridge sont classées selon leur valeur mais également selon leur couleur. L'ordre des couleurs est le suivant (ordre croissant) : trèfle, carreau, cœur, pique. L'ordre croissant sur les valeurs est : 2, 3, ..., 10, valet, dame, roi, as.

6. Écrire une fonction `carte_sup` : `carte -> carte -> bool` qui renvoie *true* si la première carte est plus forte que la seconde selon l'ordre défini ci-dessus.
7. Écrire une fonction `tri_main` : `main -> main` qui prend en entrée une main quelconque et renvoie une main rangée par ordre décroissant.
8. Écrire une fonction `dist_main` : `main -> int list` prenant en entrée une main et renvoyant sa distribution, c'est-à-dire son nombre de piques, cœurs, carreaux et trèfles sous forme d'un quadruplet d'entiers.

3 Compte des points

La valeur des cartes est donnée dans le tableau suivant (la couleur n'intervenant pas) :

	Valeur
As	4
Roi	3
Dame	2
Valet	1
2...10	0

9. Écrire une fonction `evalue_carte` : `carte -> int` qui prend en argument une carte et renvoie sa valeur.

¹david.forge@lri.fr, agrappy@limsi.fr, louis.mandel@lri.fr ou veronique.ventos@lri.fr

Évaluer une main consiste à lui associer un nombre (sa valeur). Les points d'honneur sont donnés par la somme des valeurs de chaque carte. Les points de longueurs sont donnés par carte supplémentaire à partir de la cinquième carte d'une couleur. La valeur d'une main est la somme des points d'honneur et de valeur. Par exemple, une main contenant 2 piques, 1 cœur, 6 carreaux et 4 trèfles aura 2 points de longueur provenant du cinquième et du sixième carreaux.

10. Écrire une fonction `eval_honneur_main : main -> int` permettant de compter le nombre de points d'honneur d'une main.
11. Écrire une fonction `eval_longueur_main : main -> int` permettant de compter le nombre de points de longueur d'une main.
12. Écrire une fonction `eval_main : main -> int` qui calcule la valeur totale d'une main.

Il existe une autre façon d'attribuer une valeur à une main qui calcule les points ZAR. Pour cela, on fait le total des :

- points d'honneur calculés comme précédemment mais avec les valeurs : 1 point pour un valet, 2 pour une dame, 4 pour un roi et 6 pour un As ;
- points de distribution en faisant d'une part la somme des deux longueurs les plus longues et la différence entre la longueur la plus longue et la longueur la plus courte.

Par exemple, la main composée des piques A5432, du 4 de cœur, des carreaux DV863 et des trèfles R2, s'évalue en points ZAR, par $(6 + 2 + 1 + 4) = 13$ en points d'honneur, $(5 + 5) + (5 - 1) = 14$ en points de distribution, ce qui donne un total de 27 points ZAR.

13. Écrire une fonction `compte_zar_main : main -> int` qui calcule la valeur ZAR d'une main.

4 Ouverture des mains

Avec le compte classique des points, on considère que pour « ouvrir » une main il faut qu'elle possède au moins soit 12 points d'honneur, soit 13 points totaux (honneur et distribution). En ce qui concerne le compte ZAR, on ouvre une main à partir de 26 points ZAR.

14. Écrire une fonction `ouvre_main` qui étant donnée une main renvoie un couple de booléens $(b1, b2)$ avec $b1 = true$ si on doit ouvrir classiquement la main et $false$ sinon et de même $b2 = true$ si on doit ouvrir en ZAR la main et $false$ sinon.
15. Définir la constante `jeu` qui est la liste des 52 cartes du jeu de bridge.
16. Écrire une fonction `distribue : unit -> main list` qui distribue les 52 cartes en 4 mains de manière aléatoire. On utilisera la fonction de tirage d'un entier `Random.int` qu'il faudra initialiser avec `Random.self_init()`.
17. Écrire une fonction `ouvertures_jeu : unit -> (bool * bool) list` qui renvoie pour chacune des quatre mains du jeu complet le résultat de `ouvre_main`.