ALGO5 - Travaux dirigés

Coût maximal, coût minimal. Coût moyen (analyse probabiliste)

Le coût d'une séquence d'instructions dépend en général des valeurs de certaines variables du programme. Le coût maximal (respectivement minimal) est la plus grande (respectivement plus petite) valeur du coût qui puisse **être observée** *lors d'une exécution de cette séquence. Le coût moyen est la moyenne des valeurs du coût, chacune de ces valeurs étant* **pondérée par sa probabilité d'être observée dans les conditions de l'expérience**. *Le calcul du coût moyen (sauf dans le cas trivial où le coût maximal est égal au coût minimal) nécessite donc* **toujours** *de faire des hypothèses (probabilistes) sur les valeurs de certaines variables du programme.*

Exercice 1

Soit l'algorithme suivant :

```
pour i de de 1 à n faire
        si T[i] > a alors
            s := s + T[i]
```

Evaluer le nombre d'additions exécutées par cet algorithme :
- Considérer d'abord les cas favorables (coût minimal) et défavorables (coût maximal).
- Faire l'analyse en moyenne, en prenant pour hypothèse que la probabilité pour que le test $T[i] > a$ soit *vrai* est 1/2.

Exercice 2

```
si a > b alors
    pour i = 1 à n faire
            x := x+a
sinon x := b
```

Evaluer le nombre d'additions exécutées par cet algorithme (en fonction de a et b) :
- Considérer d'abord les cas favorables (coût minimal) et défavorables (coût maximal).
- Faire l'analyse en moyenne, en prenant pour hypothèse que la probabilité pour que le test $a > b$ soit *vrai* est 1/2.
  Peut-on observer le résultat obtenu, lors d'une exécution particulière de l'algorithme ?
- Faire l'analyse en moyenne, en prenant pour hypothèse que la probabilité pour que le test $a > b$ soit *vrai* est 3/4.

Exercice 3

```
i := 1
tant que i <= n etpuis T[i] > a faire
              x := x+a
              i := i+1
```

Evaluer le nombre d'additions exécutées par cet algorithme dans les cas favorables, défavorables et en moyenne, en prenant pour hypothèse que la probabilité pour que le test $T[i] > a$ soit *vrai* est 1/2.

Exercice 4

Mêmes questions pour l'algorithme :

(1) tant que random > 1/2 faire
(2) x := x+a

Exercice 5

Mêmes questions pour l'algorithme :

(1) n := 1
(2) tant que random <= $1/n$ faire
(3) n := n+1

Exercice 6

Soit l'algorithme suivant qui calcule le maximum des éléments d'un tableau de taille n (donné en cours) :

fonction max_iteratif (tableau T, entier n)
{T est un tableau d'objets, l'ensemble des objets est muni d'une relation d'ordre total $\succ$}
max $=-\infty$;
pour i = 1 to n **faire**
  si T[i]$\succ$max **alors**
    Traiter(T[i])
    max=T[i]
  fin si
fin pour

L'objectif est d'analyser le coût de cet algorithme en nombre d'appels à la fonction **Traiter** : au mieux, au pire et en moyenne.