Université Savoie Mont Blanc, UFR SceM
INFO602, L3 INFO

5 mai 2021
Durée : 1h

Examen, INFO602, Session 1

Documents autorisés : tous documents du cours/td/tp, notes manuscrites (nb : pas de livres)

Les exercices sont indépendants. Le barème est indicatif. Il dépasse volontairement 20.

Exercice 1. Notations O, Ω, Θ (/6)

Complétez le tableau ci-dessous en indiquant si f est un grand O, un grand Ω ou un grand Θ de la fonction g (pour n tendant vers l'infini). Attention, Juste $= +0,4$, Faux $= -0,4$, Rien $= +0$, ceci pour éviter que vous ne répondiez au hasard.

		g		
		$n^2 + 3n\log n$	2^{n-1}	$3n + \frac{n}{2}\log n$
f	$n\sqrt{n} + 10$			
	$n^2(n+1)$			
	$n\log n$			
	$2^{n+1} + 2n^2$			
	$n + 3\log n$			

Exercice 2. Fonction mystère (/4)

Que calcule F ? Quelle est sa complexité en pire cas ? Plus généralement, si on remplace 10 par un entier $a \geq 2$ quelconque, quelle fonction mathématique classique approche F ?

```
// n est un nombre entier positif ou nul.
Fonction F( E n : Entier ) : entier ;
début
    if n == 0 then Retourner 0;
    else Retourner 1+F(n/10);
fin
```

Exercice 3. Complexité d'une fonction récursive (/4)

On se donne un algorithme dont le temps d'exécution est de la forme : $T(1) = O(1)$, et $T(n) = 3T(n/2) + 6n$.

Quelle est la complexité en pire cas de cet algorithme en fonction de n ? Justifiez votre résultat.

NB : il s'agit de la complexité de l'algorithme de multiplication de grands entiers sur n bits proposé par Karatsuba en 1960.

Exercice 4. Polygones simples, convexité et points intérieurs (/7,5)

Si $Q = (q_i)_{i=0..n-1}$ est un polygone simple, il existe une façon relativement simple de savoir si un point p est à l'intérieur de Q. On trace un rayon $[p, r)$ à partir de p, où r est un point différent de p, et on compte le nombre de fois où $[p, r)$ intersecte le bord de Q. Si ce nombre est impair alors le point est à l'intérieur, sinon le point est à l'extérieur (voir illustration page suivante).

Nb : dans la suite, pour simplifier, on supposera toujours que le rayon $[p, r)$ ne traverse pas un sommet de Q, ni se s'aligne avec un côté de Q. C'est une hypothèse réaliste si r est bien choisi ou si r est tiré au hasard.

Nb : on vous redonne dans la page suivante quelques fonctions de base de géométrie algorithmique, que vous pouvez utiliser dans vos algorithmes.

1. (/2) Adaptez la fonction Intersection-Segments du cours (rappelée plus loin) pour en faire une fonction Intersection-Rayon(p_1, p_2, p_3, p_4) qui retourne vrai si et seulement le rayon $[p_1, p_2)$ intersecte le segment fermé $[p_3, p_4]$.

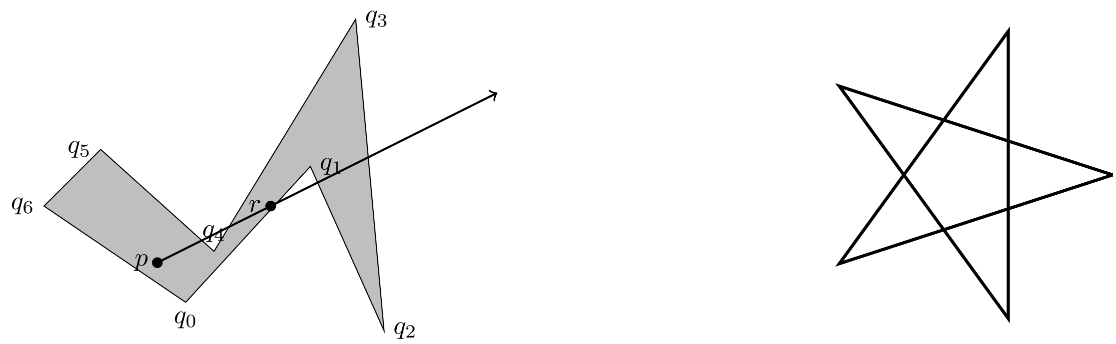

FIGURE 1 – (Gauche) le point p est dans le polygone Q car $[p, r)$ intersecte 5 fois le bord de Q. (Droite) un polygone non simple : le pentacle.

2. (/2,5) Ecrire maintenant l'algorithme qui teste si un point p est à l'intérieur de Q. On attend ici un algorithme simple de complexité linéaire en le nombre de sommets de Q. Son prototype sera **Fonction** ESTINTÉRIEUR ?*(* **E** *p : Point,* **E** *Q : Polygone) : booléen*

   NB : On écrira $Q.n$ pour avoir le nombre de sommets de Q, et $Q[i]$ pour accéder au i-ème sommet, avec l'indice i pris modulo $Q.n$. N'oubliez pas de choisir un r.
3. (/1,5) Si maintenant Q est un polygone convexe, combien de fois le rayon $[p, r)$ peut-il intersecter le bord de Q ?
4. (/1,5) Prenons maintenant Q un polygone non simple comme le pentacle. Dessinez-le et remplissez en gris les zones où cet algorithme retournera intérieur et laissez en blanc les zones où cet algorithme retournera extérieur. Y trouvez-vous une logique ?

`//` Retourne un nombre > 0 si et seulement si r est à gauche du rayon $[p, q)$, un nombre < 0 ssi r est à droite de $[p, r)$, et 0 si p, q, r sont alignés.

Fonction ORIENTATION*(* **E** *p, q, r : Point) : réel ;*

début

  Retourner $(q.x - p.x) * (r.y - p.y) - (q.y - p.y) * (r.x - p.x)$;

fin

`//` Sachant que r est sur la droite (pq), détermine si $r \in$ segment $[pq]$.

Fonction SUR-SEGMENT*(* **E** *p, q, r : Point) : booléen ;*

début

  Retourner $\min(p.x, q.x) \leq r.x \leq \max(p.x, q.x)$ *et* $\min(p.y, q.y) \leq r.y \leq \max(p.y, q.y)$;

fin

`//` Détermine si les segments $[p_1, p_2]$ et $[p_3, p_4]$ s'intersectent.

Fonction INTERSECTION-SEGMENTS*(* **E** *p_1, p_2, p_3, p_4 : Point) : booléen;*

début

  $d_1 \leftarrow$ ORIENTATION(p_3, p_4, p_1);

  $d_2 \leftarrow$ ORIENTATION(p_3, p_4, p_2);

  $d_3 \leftarrow$ ORIENTATION(p_1, p_2, p_3);

  $d_4 \leftarrow$ ORIENTATION(p_1, p_2, p_4);

  si $((d_1 < 0$ *et* $d_2 > 0)$ *ou* $(d_1 > 0$ *et* $d_2 < 0))$ *et* $((d_3 < 0$ *et* $d_4 > 0)$ *ou* $(d_3 > 0$ *et* $d_4 < 0))$

   alors Retourner *Vrai* ;

  sinon si $d_1 = 0$ *et* SUR-SEGMENT(p_3, p_4, p_1) **alors Retourner** *Vrai* ;

  sinon si $d_2 = 0$ *et* SUR-SEGMENT(p_3, p_4, p_2) **alors Retourner** *Vrai* ;

  sinon si $d_3 = 0$ *et* SUR-SEGMENT(p_1, p_2, p_3) **alors Retourner** *Vrai* ;

  sinon si $d_4 = 0$ *et* SUR-SEGMENT(p_1, p_2, p_4) **alors Retourner** *Vrai* ;

  sinon Retourner *Faux* ;

fin