

Concours blanc

Option informatique, deuxième année

Julien REICHERT

Partie I : Graphes

Soit G un graphe non orienté dont les sommets sont les entiers de 1 à 8 représenté par liste d'adjacence ci-dessous.

Sommets	Sommets adjacents
1	(2,3,4)
2	(1,3,4)
3	(1,2,4)
4	(1,2,3,6)
5	(6,7,8)
6	(4,5,7)
7	(5,6,8)
8	(5,7)

On considérera que lors du parcours du graphe les sommets adjacents d'un sommet donné sont rencontrés dans le même ordre qu'ils sont listés dans le tableau précédent.

Question 1 : Représenter graphiquement le graphe correspondant à G .

Question 2 : Ce graphe est-il complet ? Est-il connexe ?

Question 3 : Donner la séquence des sommets de G obtenus lors du parcours en profondeur en commençant sur le sommet 1.

Question 4 : Donner la séquence des sommets de G obtenus lors du parcours en largeur en commençant sur le sommet 1.

Partie II : Programmation

Question 5 : Écrire en Caml un programme d'exponentiation tel que le calcul de a^n , où les deux paramètres sont des entiers, fasse un nombre de multiplications logarithmiques en n .

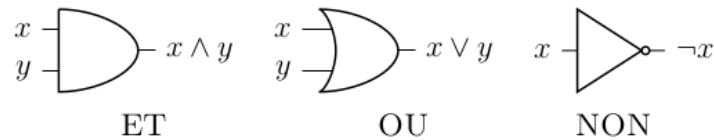
Question 6 : Écrire en Caml une fonction récursive ou faisant appel à des sous-fonctions récursives calculant la factorielle de son argument en temps linéaire en sa valeur.

Question 7 : Écrire en Caml une fonction vérifiant si la liste en argument est bitonique, c'est-à-dire qu'elle se décompose en une liste croissante suivie d'une liste décroissante, chacune éventuellement vide.¹

1. Formulation équivalente : il n'existe pas trois éléments tels que celui du milieu soit strictement inférieur aux deux autres.

Partie III : Circuits logiques

Les portes logiques sont des éléments d'un circuit dont les entrées et les sorties sont binaires. Les trois portes logiques élémentaires ET, OU et NON sont décrites ci-dessous.



Dans les questions suivantes, indiquer clairement les entrées et les sorties dans tous les diagrammes de circuit. Éviter autant que possible d'utiliser des symboles supplémentaires et si c'est le cas, indiquer dans votre réponse ce qu'ils représentent.

Question 8 : En utilisant uniquement les portes ET, OU et NON, dessiner un diagramme de circuit pour un additionneur de 1 bit qui prend en entrée deux bits a_0 , b_0 et (éventuellement) une retenue r et retourne la somme s_0 de a_0 , b_0 et r et la retenue s_1 , dont la formule mathématique est rappelée ci-dessous.

$$s_0 = a_0 + b_0 + r \bmod 2 \text{ et } s_1 = \lfloor (a_0 + b_0 + r)/2 \rfloor$$

Question 9 : En utilisant uniquement les portes ET, OU et NON et le circuit de la question précédente, dessiner un diagramme pour un additionneur de 3 bits qui prend en entrée a_2 , b_2 , a_1 , b_1 , a_0 , b_0 , r représentant deux nombres de 3 bits a and b en binaire (notés $a_2a_1a_0$ et $b_2b_1b_0$), et un bit de retenue r . Il devra retourner la somme s de ces trois nombres en binaire sous la forme $s_3s_2s_1s_0$.

Question 10 : Expliquer comment étendre l'additionneur de la question précédente en un additionneur pour n bits et déterminer le nombre de portes nécessaires en fonction de n .

Question 11 : Supposons que le temps de propagation d'un signal logique à travers une porte élémentaire est de t secondes (indépendamment de la porte). Si tous les signaux d'entrée sont altérés pour l'additionneur pour n bits de la question précédente au temps 0, quel est le temps nécessaire pour l'obtention du résultat en sortie du circuit (en fonction de n et t) ?

Nous appelons un additionneur de 2^k bits « diviser pour régner » le circuit défini récursivement de la façon suivante :

- L'additionneur de 1 bit « diviser pour régner » est l'additionneur de 1 bit de la première question.
- Pour $k \geq 1$, l'additionneur de 2^k bits « diviser pour régner » est obtenu en fusionnant deux additionneurs de 2^{k-1} bits « diviser pour régner » A et B en envoyant les 2^{k-1} bits de poids faible des entrées a et b et l'éventuelle retenue r à A et tous les bits restants à B . La retenue de la sortie de A est envoyée à B .

Question 12 : Déduire le délai pour l'obtention du résultat pour ce circuit en fonction de $n = 2^k$ et de t .

La fin du sujet n'est pas évaluée mais est donnée pour le plaisir.

Nous considérons maintenant un additionneur de 2^k bits « diviser pour régner » amélioré où le circuit agissant sur les bits de poids forts est remplacé par deux additionneurs, l'un supposant que la retenue entrante vaut 1 et l'autre supposant que la retenue entrante vaut 0.

- Dessiner le diagramme de circuit utilisant trois additionneurs de 2^{k-1} bits « diviser pour régner » amélioré et quelques portes logiques élémentaires pour construire un additionneur de 2^k bits « diviser pour régner » amélioré.
- Déduire le délai pour l'obtention du résultat pour cet additionneur de 2^k bits « diviser pour régner » amélioré en fonction de $n = 2^k$ et de t .
- Donner une formule récursive donnant le nombre de portes logiques élémentaires dans cet additionneur.