

Circuits. PC 3.

1 Parité

La fonction PARITY_n est la fonction $\{0, 1\}^n \rightarrow \{0, 1\}$ qui vaut 1 si et seulement si un nombre impair de ses arguments valent 1. Montrer que PARITY_n se calcule par un circuit booléen de profondeur $\mathcal{O}(\log(n))$ et de taille $\mathcal{O}(n)$.

2 Majorité

1. Soit $\text{ADD}_n : \{0, 1\}^n \rightarrow \{0, 1\}^{n+1}$ la fonction qui prend en entrée deux nombres binaires et produit le résultat de leur somme. Montrer qu'elle peut se calculer par un circuit booléen de de taille $\mathcal{O}(n)$.
2. On définit $\text{MAJORITY}_n : \{0, 1\}^n \rightarrow \{0, 1\}$ telle que $\text{MAJORITY}_n(x_1, \dots, x_n)$ vaut 0 si $\sum x_i < n/2$, 1 sinon. Montrer que cette fonction se calcule
 - (a) par un circuit de taille $\mathcal{O}(n^2)$.
 - (b) par un circuit de taille $\mathcal{O}(n \log(n))$.
 - (c) par un circuit de taille $\mathcal{O}(n)$ (on pourra utiliser le fait que l'addition de 3 entiers se ramène à calculer la somme de deux entiers en profondeur constante).

3 Réseaux de tri

Considérons une liste $L = x_0, x_1, \dots, x_{n-1}$ de n éléments de E , un ensemble ordonné, avec n puissance de 2.

La liste L est dite *bitonique* si L est dans l'un des quatre cas suivants :

- Cas 1 : L est croissante.
- Cas 2 : L est décroissante.
- Cas 3 : L est croissante puis décroissante.
- ou un décalage circulaire de la liste est dans l'un de ces cas.

Voici un exemple de liste bitonique : 10 20 30 40 50 60 70 80 75 65 55 45 35 25 15 5. Un autre exemple 6 9 4 2 3 5.

Un *circuit comparateur* est un circuit $C(x, y)$ à deux entrées x et y et deux sorties : la première donne le minimum de x et de y , et la seconde le maximum de x et de y .

On appelle *réseau de tri* un circuit constitué de circuits comparateurs.

On appelle un *réseau séparateur* un réseau de tri, constitué d'une ligne de circuits comparateurs tel que si la liste $L = x_0, x_1, \dots, x_{n-1}$ est donnée en entrée (avec $n = 2p$), alors seuls les éléments x_i et x_{i+p} sont comparés ensemble.

1. Montrer qu'un réseau séparateur appliqué à une liste bitonique produit une liste telle que :
 - (a) tout élément de la première moitié de la liste de sortie est inférieur à tous les éléments de la seconde moitié de la liste de sortie.
 - (b) la première moitié et la seconde moitié de la liste de sortie sont des listes bitoniques de $p = n/2$ éléments.
2. Construire un circuit constitué de circuits comparateurs qui trie les éléments d'une liste bitonique en utilisant des réseaux séparateur. Évaluer la taille et la profondeur du circuit.
3. Expliquez comment adapter le réseau *séparateur* pour trier en ordre décroissant une liste bitonique.
4. Décrire un réseau *fusion* qui transforme une liste en une liste bitonique en utilisant les réseaux tri croissant et tri décroissant.
5. En déduire un circuit qui trie les éléments d'une liste quelconque constitué uniquement de circuits comparateurs avec une profondeur $\mathcal{O}(\log(n)^2)$.
6. Montrer que tout réseau de tri à n entrée possède $\Omega(n \log n)$ circuits comparateurs.
7. Montrer que tout réseau de tri à n entrée possède une profondeur au moins $\log(n)$ (voir votre cours préféré).

4 Circuits sur \mathbb{R}

On considère des circuits et algorithmes sur la structure $(\mathbb{R}, +, -, *, =, <)$.

On cherche, étant donnés n nombres x_1, x_2, \dots, x_n à déterminer s'ils sont tous distincts deux à deux (problème DISTINCT)

1. Proposer un algorithme en temps $\mathcal{O}(n \log n)$.
2. Lorsque $W \subset \mathbb{R}^n$, on note $\#(W)$ pour le nombre de ses composantes connexes.
Soit $W = \{(x_1, \dots, x_n) \mid \prod_{i \neq j} (x_i - x_j) \neq 0\}$. Montrer que $\#(W) \geq n!$.
3. Montrer que si C est un arbre de décision sur $(\mathbb{R}, +, -, *, =, <)$ à n entrées de profondeur d , alors l'ensemble des entrées acceptés est une union d'au plus 2^d ensembles $C_1, C_2, \dots, \subset \mathbb{R}^n$, où chaque C_i se définit par un système de $\leq d$ équations ou inéquations polynomiales de degré 2.
4. On admettra que si $S \subset \mathbb{R}^n$ est défini par des contraintes de degré d avec m égalités et h inégalités, alors $\#(S) \leq d(2d-1)^{n+h-1}$ (ne dépend pas de m).
En déduire qu'il faut au moins un temps $\mathcal{O}(n \log n)$ pour résoudre le problème DISTINCT.