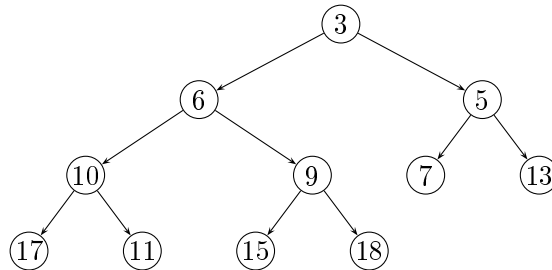


□ **Exercice 1** : *Insertion dans un tas*

1. Rappeler la définition d'un tas binaire (min)
2. On suppose qu'un tas est représenté par un tableau $t = (t_0, \dots, t_{n-1})$. Lorsqu'ils existent quels sont les indices des fils de t_i ?
3. Quel est l'indice (lorsqu'il existe) du père de t_i ?
4. Rappeler le principe d'insertion d'un nouvel élément dans un tas binaire
5. Vérifier que l'arbre binaire suivant possède bien la structure de tas :



6. Détailler les étapes de l'insertion de 4 dans le tas précédent.
7. On rappelle les structures de données vues en cours et permettant de représenter un tas :
 - En langage C :

```

1 struct heap
2 {
3     int size;
4     int capacity;
5     int *tab;
6 };
7 typedef struct heap heap;

```

— En OCaml :

```

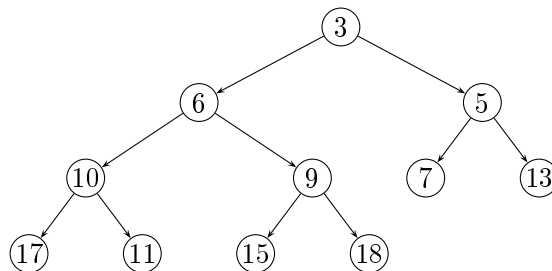
1 type heap_int = {mutable size : int; data : int array};;

```

Dans le langage de votre choix, implémenter l'algorithme d'insertion dans un tas.

□ **Exercice 2** : *Extraction du minimum dans un tas*

1. Rappeler la définition d'un tas binaire (min)
2. On suppose qu'un tas est représenté par un tableau $t = (t_0, \dots, t_{n-1})$. Lorsqu'ils existent quels sont les indices des fils de t_i ?
3. Quel est l'indice (lorsqu'il existe) du père de t_i ?
4. Rappeler le principe d'extraction du minimum d'un tas binaire
5. Vérifier que l'arbre binaire suivant possède bien la structure de tas :



6. Détailler les étapes de l'extraction du minimum dans le tas précédent.
7. On rappelle les structures de données vues en cours et permettant de représenter un tas :
 - En langage C :

```

1 struct heap
2 {
3     int size;
4     int capacity;
5     int *tab;
6 };
7 typedef struct heap heap;

```

— En OCaml :

```

1 type heap_int = {mutable size : int; data : int array};;

```

Dans le langage de votre choix, implémenter l'algorithme d'extraction du minimum dans un tas.

□ Exercice 3 : Ensemble inductif

Soit X l'ensemble construit par induction à partir :

- des axiomes $X_0 = 0$
- des règles d'inférence $r_1 : n \mapsto n + 5$, $r_2 : n \mapsto n + 2$ et $r_3 : n \mapsto -n$
 1. En donnant une suite de règle permettant de l'obtenir, montrer que $11 \in X$.
 2. La définition de X est-elle ambiguë ?
 3. Prouver par induction structurelle que $X \subset \mathbb{Z}$.
 4. Montrer que $X = \mathbb{Z}$.
 5. On définit en OCaml le type :

```

1 type nb =
2 | Z
3 | R1 of nb
4 | R2 of nb
5 | R3 of nb;;

```

Ecrire une fonction `res nb -> int` qui prend en renvoie l'entier associée à la suite d'applications des règles d'inférences. Par exemple `res (R3 (R1 (R2 Z)))` renvoie `-7`.

□ Exercice 4 : Recherche d'un pic par la méthode diviser pour régner

On dit qu'un tableau t de taille n présente un *pic* en position p si et seulement si :

- toutes les valeurs de t sont distinctes
 - le sous tableau t_0, \dots, t_p est trié par ordre croissant
 - le sous tableau t_p, \dots, t_{n-1} est trié par ordre décroissant.
1. Justifier rapidement que si un tableau présente un pic alors ce pic est unique.
 2. Ecrire dans le langage de votre choix, une fonction itérative de complexité linéaire qui prend en entrée un tableau présentant un pic et renvoyant la position de ce pic.
 3. On veut maintenant utiliser une méthode diviser pour régner.
 - a) Donner le cas de base et sa solution
 - b) Si on divise le tableau en deux moitiés, montrer qu'on peut déterminer dans laquelle se trouve le pic.
 - c) Donner l'implémentation de la méthode diviser pour régner dans le langage de votre choix.
 - d) Donner sa complexité.