

問題3 次の木構造に関する記述を読み、各設問に答えよ。

木構造とは、一つの要素（節：node）から枝のようにいくつかの子要素を持つデータ構造で、子要素はさらに子要素を持つことができるため、階層的なデータ構造として使われる。親の無い節を根（root）と呼び、子要素を持たない節を葉（leaf）と呼ぶ。なお、ここで木構造に格納する要素の値は0以上の整数とする。

<設問1> 次の2分木に関する記述中の□に入るべき適切な字句を解答群から選べ。

一つの節が持つ子要素の数が二つ以下である木構造のことを2分木（binary tree）と呼び、次の条件が常に成立するような構造になっている。

[条件] (左側の子要素の値) ≤ (親の値) ≤ (右側の子要素の値)

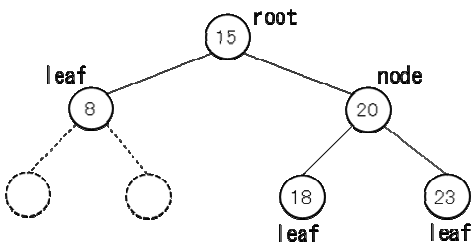


図1 2分木の例

図1の2分木の例を1次元配列BTで表現する。

| | | | | | | | | | | | | | | |
|-------|----|---|---|----|----|---|----|----|----|----|----|----|----|-----|
| 添字 i | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | ... |
| 配列 BT | 15 | 8 | X | -1 | -1 | Y | 23 | -1 | -1 | -1 | -1 | -1 | -1 | ... |

図2 1次元配列BT

- ・根は、BT[1]に格納される。
- ・どの要素位置に対しても、添字 i に対して、左側の子要素は BT[2i] に、右側の子要素は BT[2i+1] に格納される。なお、未使用領域には-1を格納する。
- ・配列 BT は、処理に必要な十分な大きさを持っているものとする。

図1の2分木を上記の方法で1次元配列BTを用いて表現すると、Xには□(1)が入り、Yには□(2)が入る。

また、図2の配列BTに10を追加した場合、配列BT[□(3)]に追加され、追加後の葉（leaf）の数は□(4)個となる。

(1) ～ (4) の解答群

- ア. -1 イ. 3 ウ. 4
エ. 5 オ. 18 カ. 20

<設問 2> 次のヒープに関する記述中の に入るべき適切な字句を解答群から選べ。

節の値はその節のどの子よりも小さい（または大きい）2分木をヒープと呼ぶ。なお、ヒープでは、葉は左詰めにし、子要素どうしの大小関係は問わない。ここで扱うヒープは、次の条件が常に成立するような構造になっている。

[条件] (親の値) \leq (全ての子の値)

次のデータをヒープ構造で表現すると図3のようになる。ここで、P, Q, Rに入る組み合わせは (5) となる。

[データ] 8, 14, 15, 20, 23

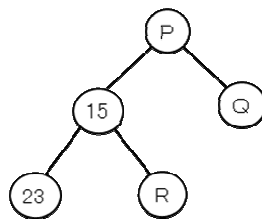


図3 ヒープの例

(5) の解答群

| | P | Q | R |
|----|----|----|----|
| ア. | 8 | 14 | 20 |
| イ. | 14 | 8 | 20 |
| ウ. | 14 | 20 | 8 |
| エ. | 20 | 14 | 8 |

図4のヒープ構造を持つデータから4を削除した場合、根 (root) の位置に格納されるデータは (6) となる。

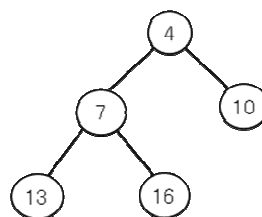


図4

(6) の解答群

ア. 7

イ. 10

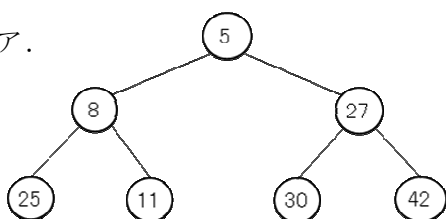
ウ. 13

エ. 16

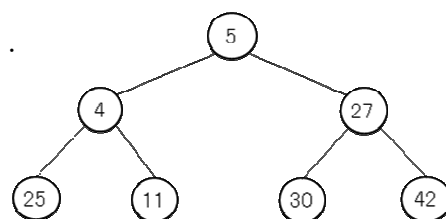
次の 2 分木データで、ヒープの条件を満たしているものは (7) である。

(7) の解答群

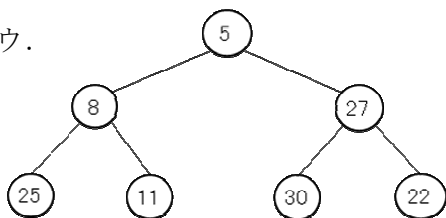
ア.



イ.



ウ.



エ.

