

I - Alea jacta est!

题目

题目描述

有 2^n 只猫参加晋级赛，这些猫依次编号为 $1, 2, \dots, 2^n$ 。编号为 i ($1 \leq i \leq 2^n$) 的猫有实力值 p_i ，每只猫的实力值互不相同。

晋级赛分为 n 轮，第 i ($1 \leq i \leq n$) 轮将有 2^{n-i+1} 只猫参加。这意味着在第 i 轮中将有 2^{n-i} 只猫被淘汰，第 n 轮结束后的仍在赛场上的猫获胜。

具体来说，在第 i 轮比赛中将会依次发生如下事件：

- 仍在赛场上的猫按照其编号从小到大排成一列；
- 对于 $1 \leq k \leq 2^{n-i}$ ，编号第 $2k - 1$ 小的猫与编号第 $2k$ 小的猫互为对手，二者中实力值较小者被淘汰；
- 若仍有超过一只未被淘汰的猫，则这些猫进入第 $i + 1$ 轮比赛。

现在每只猫想知道，它在比赛中所遭遇的所有对手中，实力值最大是多少。猫的数量过多，它们自己实在是难以计算，所以需要你的帮助。

输入

本题测试点包含多组数据。

第一行，一个正整数 T ($1 \leq T \leq 10$)，表示数据组数。

对于每组数据：

第一行，一个正整数 n ($1 \leq n \leq 18$)，表示晋级赛轮数。

第二行， 2^n 个整数 p_i ($0 \leq p_i \leq 10^9$)， p_i 表示编号为 i 的猫的实力值。

对于所有数据，保证有 $\sum 2^n \leq 2^{20}$ 。

输出

对于每组数据：

输出一行， 2^n 个整数，第 i 个整数表示编号为 i 的猫所遭遇的所有对手的最大实力值。

题意

给出 2^n 只🐱的能力值，从左往右相邻地两两比较，比较n轮后决出冠军，要输出每只🐱遇到过的最大的能力值。

思路

暴力模拟

一个数组`p`记录所有🐱的能力值，另一个数组`ans`记录所有🐱遇到过的最大的能力值，即最后输出的答案。

对能力值数组进行`n`轮遍历，将每次比较中输掉的🐱的能力值设为`-1`，并不断更新`ans`数组。在每一轮遍历比较时跳过能力值为`-1`的项，因此需要判断相邻两个能力值为`c1 c2`，`c1 -1`，`-1 c2`，`-1 -1`四种情况。

代码

```
#include <stdio.h>
#define MAXN (1 << 18) + 5
int p[MAXN], ans[MAXN];

void function(int n, int m) {
    int k = 0;
    while (n) { //进行n轮比较

        for (int i = 1; i <= m; i++) { //遍历每只🐱的能力值
            if (p[i] != -1 && p[i + 1] != -1) { //1.未淘汰 未淘汰, i后移两位
                if (p[i + 1] > ans[i])
                    ans[i] = p[i + 1];
                if (p[i] > ans[i + 1])
                    ans[i + 1] = p[i];

                if (p[i] > p[i + 1])
                    p[i + 1] = -1;
                else
                    p[i] = -1;

                i += 2;
            }
            else if (p[i] == -1 && p[i + 1] != -1) //2.已淘汰 未淘汰, i后移一位
                i++;
            else if (p[i] != -1 && p[i + 1] == -1) { //3.未淘汰 已淘汰 j记录下未淘汰的位置, i不断后移直到
                //找到未淘汰的🐱
                int j = i;
                i++;
                while (p[i] == -1)
                    i++;

                if (p[i] > ans[j])
                    ans[j] = p[i];
                if (p[j] > ans[i])
                    ans[i] = p[j];

                if (p[j] > p[i])
                    p[i] = -1;
                else
                    p[j] = -1;

                i++;
            }
            else if (p[i] == -1 && p[i + 1] == -1) //4.已淘汰 已淘汰, i后移一位
                i++;
        }
        n--;
    }
}
```

```
        }
        n--;    //一轮比较完, n--
    }
}

int main() {
    int t;
    scanf("%d", &t);
    while (t-->0) {
        int n;
        scanf("%d", &n);
        int m = 1 << n;
        for (int i = 1; i <= m; i++) {
            scanf("%d", &p[i]);
            ans[i] = 0;
        }
        function(n, m);
        for (int i = 1; i < m; i++)
            printf("%d ", ans[i]);
        printf("%d\n", ans[m]);
    }
    return 0;
}
```