

1-1 分数 2

作者 叶德仕 单位 浙江大学

The  $n$ -th Fibonacci number can be computed by divide and conquer method of computing  $x^n$ , where  $x$  is the matrix

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

Then the  $n^2$ -th Fibonacci number  $F_{n^2}$  can be computed in  $O(\log n)$  time.

☒ T ☐ F

评测结果 答案正确

得分 2 分

1-2 分数 2

作者 何钦铭 单位 浙江大学

When insert three keys into a non-empty 2-3 tree, and if the tree gains height when the first key is in, then we can be sure that the 2-3 tree will **not** gain height after the insertions of the next two keys.

☐ T ☒ F

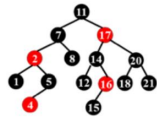
评测结果 答案错误

得分 0 分

1-3 分数 2

作者 杨洋 单位 浙江大学

The following binary search tree is a valid red-black tree.



☒ T ☐ F

评测结果 答案错误

得分 0 分

1-4 分数 2

作者 陈越 单位 浙江大学

The number of light nodes along the right path of a skew heap is  $O(\log N)$ .

☒ T ☐ F

评测结果 答案正确

得分 2 分

1-5 分数 2

作者 徐镜春 单位 浙江大学

For the recurrence equation  $T(N) = 9T(N/3) + N^2 \log N$ , we obtain  $T(N) = O(N^2 \log N)$  according to the Master Theorem.

☐ T ☒ F

评测结果 答案正确

得分 2 分

1-6 分数 2

作者 陈越 单位 浙江大学

In order to prove the amortized time bound for a skew heap, the potential function can be defined to be the number of right nodes of the resulting tree.

☐ T ☒ F

评测结果 答案正确

得分 2 分

2-1 分数 3

作者 陈越 单位 浙江大学

Starting from the red-black tree given in the figure, after successively inserting the keys {75, 70, 50}, which one of the following statements is FALSE?



- ☐ A. 50 is the deepest red node
- ☒ B. 70 and 99 are siblings, and they are both black
- ☐ C. 70 is the parent of 75
- ☐ D. there are two red nodes

评测结果 答案正确

得分 3 分

2-2 分数 3

作者 Yang Yang 单位 浙江大学

In the context of Divide and Conquer algorithms, which of the following statements is true regarding the "Merge Sort" algorithm?

- ☒ A. Merge Sort has a average-case time complexity of  $O(n \log n)$  and uses a divide-and-conquer approach to sort a list of elements.
- ☐ B. Merge Sort has a worst-case time complexity of  $O(n^2)$  and employs a greedy strategy to sort elements efficiently.
- ☐ C. Merge Sort is not suitable for sorting large datasets due to its inefficient memory usage.
- ☐ D. Merge Sort performs sorting by repeatedly partitioning the array into subarrays and selecting pivot elements for comparison.

评测结果 答案正确

得分 3 分

2-3 分数 3

作者 Yang Yang 单位 浙江大学

In a binomial queue, we denote the total number of the nodes at even depth and odd depth as  $N_1$  and  $N_2$ , respectively (the root is defined to be at the depth 0). Which of the following statements is FALSE?

- ☒ A. If  $N_1 > N_2$ , then  $N_1 + N_2$  can be even.
- ☐ B. If  $N_1 + N_2$  is odd, then  $N_1 > N_2$ .
- ☐ C. For all cases,  $N_1 \geq N_2$ .
- ☐ D. For all cases,  $N_1 - N_2 \leq 1$ .

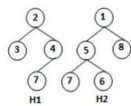
评测结果 答案正确

得分 3 分

2-4 分数 3

作者 陈越 单位 浙江大学

Merge the two leftist heaps in the following figure. Which one of the following statements is FALSE?



- ☐ A. 4 and 3 are siblings
- ☒ B. along the left path from the root, we have 1, 2, 4, and 8
- ☐ C. 6 is the right child of 5
- ☐ D. 2 and 4 have the same NPL

评测结果 答案正确

得分 3 分

2-5 分数 3

作者 陈越 单位 浙江大学

Insert  $\{3, 1, 4, 5, 0, 9, 2, 6, 8, 7\}$  into an initially empty 2-3 tree (with splitting). Which one of the following statements is FALSE?

- ☐ A. 6 and 7 are in the same node
- ☐ B. the parent of the node containing 5 has 3 children
- ☒ C. the first key stored in the root is 6
- ☐ D. there are 5 leaf nodes

评测结果 答案正确

得分 3 分

2-6 分数 3

作者 Yang Yang 单位 浙江大学

We all know how to use the binary search method to find a value in a monotonic array. As an extension, we will use the ternary search method to find the extremum in a quadratic-function-like array. Formally, for an array  $a$ , let  $k$  be the position of its extremum. Then we have either  $a[1] < \dots < a[k-1] < a[k] > a[k+1] > \dots > a[n]$  or  $a[1] > \dots > a[k-1] > a[k] < a[k+1] < \dots < a[n]$ . Apply the ternary search method to locate  $k$ . For a given interval  $1 \leq l < r \leq n$ , locate the two trisection points in  $[l, r]$  as  $m1 = l + \lfloor \frac{r-l}{3} \rfloor$  and  $m2 = l + \lfloor \frac{2(r-l)}{3} \rfloor$ . Assuming we are looking for the minimum value, which of the following pieces of code can correctly narrow down the target interval?

- ☐ A. if  $(a[m1] < a[m2])$   $l = m1$ ; else  $r = m2$ ;
- ☐ B. if  $(a[m1] < a[m2])$   $l = m2$ ; else  $r = m1$ ;
- ☒ C. if  $(a[m1] < a[m2])$   $r = m2$ ; else  $l = m1$ ;
- ☐ D. if  $(a[m1] < a[m2])$   $r = m1$ ; else  $l = m2$ ;

评测结果 答案正确

得分 3 分

奶牛渡河

时间限制: 1 Sec 内存限制: 128 MB

题目描述

Farmer John以及他的N(1 <= N <= 2,500)头奶牛打算过一条河，但他们所有的渡河工具， 仅仅是一个木筏。  
由于奶牛不会划船， 在整个渡河过程中， FJ必须始终在木筏上。 在这个基础上， 木筏上的奶牛数目每增加1， FJ把木筏划到对岸就得花更多的时间。  
当FJ一个人坐在木筏上， 他把木筏划到对岸需要M(1 <= M <= 1000)分钟。 当木筏搭载的奶牛数目从i-1增加到i时， FJ得多花M<sub>i</sub>(1 <= M<sub>i</sub> <= 1000)分钟才能把木筏划过河（也就是说， 船上有1头奶牛时， FJ得花M+M<sub>1</sub>分钟渡河； 船上有2头奶牛时， 时间就变成M+M<sub>1</sub>+M<sub>2</sub>分钟。 后面 的依此类推）。 那么， FJ最少要花多少时间， 才能把所有奶牛带到对岸呢？ 当然， 这个时间得包括FJ一个人把木筏从对岸划回来接下一批的奶牛的时间。

输入：

第1行: 2个用空格隔开的整数: N 和 M

第2..N+1行: 第i+1为1个整数: M<sub>i</sub>

输出：

第1行: 输出1个整数， 为FJ把所有奶牛都载过河所需的最少时间

样例输入：

5 10  
3  
4  
6  
100  
1

样例输出：

50

提示：

【输入说明】 FJ带了5头奶牛出门。如果是单独把木筏划过河， FJ需要花10分钟， 带上1头奶牛的话， 是13分钟， 2头奶牛是17分钟， 3头是23分钟， 4头是123分钟， 将5头一次性载过去， 花费的时间是124分钟。  
【输出说明】 Farmer John第一次带3头奶牛过河（23分钟）， 然后一个人划回来（10分钟）， 最后带剩下的2头奶牛一起过河（17分钟）， 总共花费的时间是23+10+17 = 50分钟。

```
#include<bits/stdc++.h>
using namespace std;
int dp[2550];
int main()
{
    int n,m;
    cin>>n>>m;
    dp[0]= 10    1分 ;
    for(int i=1;i<=n;i++)
    {
        int x;
        cin>>x;
        dp[i] = dp[i-1] + x;    2分
    }
    for(int i=1;i<=n;i++)
    {
        for(int j=1;j<=i;j++)
        {
            dp[i]=min(dp[i], dp[j] + dp[i-j] + 10    2分 );
        }
    }
    cout<<dp[n]<<endl;
    return 0;
}
```

评测结果 部分正确

得分 2 分

5-2 分数 5 棋盘覆盖

作者 Yang Yang 单位 浙江大学

给出一个 $2^k \times 2^k$  ( $0 < k \leq 10$ )的棋盘, 棋盘内有且仅有一个格子( $x_0, y_0$ )是特殊方格. 现要求使用总方格数为3的L型模式板覆盖棋盘除了特殊方格的其它方格, 要求模式板之间不重叠. 输出任意一种合法的覆盖方案.

输入样例：

2  
1 1

其中第一行为k, 第二行依次为x0, y0。

输出样例：

2 2 5 5  
2 0 1 5  
4 1 1 3  
4 4 3 3

其中0代表特殊方块, 1至5分别代表5个模式板。

```

#include<stdio.h>
#define N 1024

int ans[N][N];
int cnt=1;

void dfs(int x,int y,int x0,int y0,int s) {
    if(s==0) return;
    int cur=cnt++;
    int d=s-1, n=1<<d;
    int flagx = (x0<n), flagy = (y0<n);
    int loc = 2*flagx + flagy 2分;

    dfs( X,y,flagx,flagy 3分, s-1);
    if(loc==0) {ans[x+n][y+n]=cur; dfs(x+n, y+n, 0, 0, s-1);}
    if(loc==1) {ans[x+n][y+n-1]=cur; dfs(x+n, y, 0, n-1, s-1);}
    if(loc==2) {ans[x+n-1][y+n]=cur; dfs(x, y+n, n-1, 0, s-1);}
    if(loc==3) {ans[x+n-1][y+n-1]=cur; dfs(x, y, n-1, n-1, s-1);}
    return;
}

int main() {
    int k, x0, y0;
    scanf("%d%d%d", &k, &x0, &y0);
    dfs(0,0,x0,y0,k);
    int n = 1<<k;
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            printf("%d ", ans[i][j]);
        }
        puts("");
    }
    return 0;
}

```

评测结果 部分正确

得分 2分