2-1 分数 2 作者 杨欣豫 单位 浙江大学

In a binomial queue with 100 nodes, how many nodes have depth 1 (the root has depth 0)?

- A. 6
- B. 13
- C. 20
- D. Cannot be determined

2-2 分数 2 作者 何钦铭 单位 浙江大学

After deleting number 14 from a binomial queue of 5 numbers { 12, 13, 14, 23, 24 }, which of the followings is impossible?

- A. the LeftChild link of the node 12 is NULL;
- B. the NextSibling link of the node 12 is NULL;
- C. the NextSibling link of node 13 may point to node 23;
- D. the LeftChild link of node 24 is NULL;

2-3 分数 3 作者 刘明锐 单位 浙江大学

In a binomial queue, the total number of the nodes at even depth is always ___ than that of the nodes at odd depth (the root is defined to be at the depth 0).

- A. not smaller
- B. not larger
- C. smaller
- D. larger

2-4 分数 2 作者 何钦铭 单位 浙江大学

The turnpike reconstruction problem is to reconstruct a point set from distances between every pair of points. Given a set of distances {2, 2, 3, 3, 4, 5, 6, 7, 8, 10}, there are 5 corresponding points. Assume that X1 is at 0 and X5 is at 10. Which of the following statements is **TRUE**?

○ A. X2=3, X3=6, X4=8

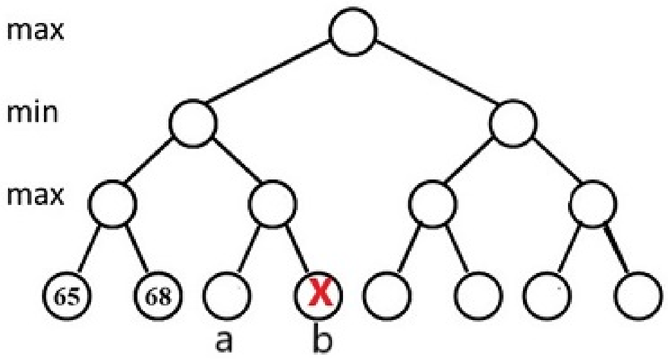
○ B. X2=3, X3=4, X4=8

○ C. X2=2, X3=4, X4=8

○ D. X2=2, X3=6, X4=7

2-5 分数 2 作者 何钦铭 单位 浙江大学

Given the following game tree, if node **b** is pruned with α-β pruning algorithm, which of the following statements about the value of node **a** is correct?



○ A. greater than 65

○ B. less than 65

○ C. greater than 68

○ D. less than 68

2-6 分数 3 作者 杨欣豫 单位 浙江大学

When solving a problem with input size N by divide and conquer, if at each step, the problem is divided into 9 sub-problems and each size of these sub-problems is $N/3$, and they are conquered in $O(N^2 logN)$. Which one of the following is the closest to the overall time complexity?

○ A. $O(N^2 log^2 N)$

○ B. $O(N^2 logN)$

○ C. $O(N^2)$

○ D. $O(N^3 logN)$

2-7 分数 2 作者 徐镜春 单位 浙江大学

Suppose that the devide-and-conquer strategy is used to find the maximum and the minimum of N positive numbers. At each step, the problem is divided into 2 sub-problems of size $N/2$. Then the time recurrences is $T(N) = 2T(N/2) + f(N)$, where $f(N)$ is ___.

○ A. $\Omega(N)$

○ B. $O(1)$

○ C. $N/2$

○ D. $\Theta(logN)$

2-8 分数 3 作者 叶德仕 单位 浙江大学

Which of the asymptotic upper bound for the following recursive $T(n)$ is correct?

○ A. $T(n) = 2T(n/2) + n\log^2 n$. Then $T(n) = O(n\log^2 n)$.

○ B. $T(n) = T(n^{1/3}) + T(n^{2/3}) + \log n$. Then $T(n) = O(\log n \log\log n)$

○ C. $T(n) = 3T(n/2) + n$. Then $T(n) = O(n)$.

○ D. $T(n) = 2T(\sqrt{n}) + \log n$. Then $T(n) = O(\log n)$.

2-9 分数 3 作者 叶德仕 单位 浙江大学

Consider two disjoint sorted arrays $A[1 \dots m]$ and $B[1 \dots n]$, we would like to compute the k-th smallest element in the union of the two arrays, where $k \le \min\{m, n\}$. Please choose the smallest possible running time among the following options.

○ A. $O(\log k)$

○ B. $O(\log m)$

○ C. $O(\log n)$

○ D. $O(\log m + \log n)$

2-10 分数 2 作者 叶德仕 单位 浙江大学

To solve the optimal binary search tree problem, we have the recursive equation $c_{ij} = \min_{i \le l \le j}\{w_{ij} + c_{i,l-1} + c_{l+1,j}\}$. To solve this equation in an iterative way, we must fill up a table as follows:

○ A.
```
for i= 1 to n-1 do;
  for j= i to n do;
    for l= i to j do
```

○ B.
```
for j= 1 to n-1 do;
   for i= 1 to j do;
     for l= i to j do
```

○ C.
```
for k= 1 to n-1 do;
  for i= 1 to n-k do;
     set j = i+k;
     for l= i to j do
```

○ D.
```
for k= 1 to n-1 do;
  for i= 1 to n do;
     set j = i+k;
     for l= i to j do
```

2-11 分数 1 作者 叶德仕 单位 浙江大学

Which one of the following problems can be best solved by dynamic programming?

○ A. Mergesort

○ B. Closest pair of points problem

○ C. Quicksort

○ D. Longest common subsequence problem

2-12 分数 2 作者 陈越 单位 浙江大学

When solving the problem All-Pairs Shortest Path by Floyd method, which one of the following iterations can give us the correct answer?

○ A.
```
for( i = 0; i < N; i++ )
    for( k = 0; k < N; k++ )
        for( j = 0; j < N; j++ )
            if( D[ i ][ k ] + D[ k ][ j ] < D[ i ][ j ] )
                D[ i ][ j ] = D[ i ][ k ] + D[ k ][ j ];
```

B.
```
for( i = 0; i < N; i++ )
    for( j = 0; j < N; j++ )
        for( k = 0; k < N; k++ )
            if( D[ i ][ k ] + D[ k ][ j ] < D[ i ][ j ] )
                D[ i ][ j ] = D[ i ][ k ] + D[ k ][ j ];
```

C.
```
for( i = 0; i < N; i++ )
    for( k = 0; k < N; k++ )
        for( j = 0; j < N; j++ )
            if( D[ k ][ i ] + D[ i ][ j ] < D[ k ][ j ] )
                D[ k ][ j ] = D[ k ][ i ] + D[ i ][ j ];
```

D.
```
for( k = 0; k < N; k++ )
    for( i = 0; i < N; i++ )
        for( j = 0; j < N; j++ )
            if( D[ k ][ i ] + D[ i ][ j ] < D[ k ][ j ] )
                D[ k ][ j ] = D[ k ][ i ] + D[ i ][ j ];
```