

浙江大学2018-19秋冬《数据结构基础》期末模拟练习

开始时间 1/1/2016, 12:00:00 AM 结束时间 1/18/2038, 12:00:00 AM 答题时长 120分钟
考生 孙恺元 得分 89 总分 100

判断题

总分：20 得分：18

1-1 If N numbers are stored in a doubly linked list in increasing order, then the average time complexity for binary search is $O(\log N)$. (2分)

☐ T ☒ F

评测结果：答案正确 (2 分)

1-2 To sort N records by quick sort, the worst-case time complexity is $O(N \log N)$. (2分)

☐ T ☒ F

评测结果：答案正确 (2 分)

1-3 For a graph, if each vertex has an even degree, we can find an Euler circuit that visits every vertex exactly once. (2分)

☐ T ☒ F

评测结果：答案正确 (2 分)

1-4 Given a binary search tree with 20 integer keys which include 5, 6, and 7, if 5 and 7 are on the same level, then 6 must be their parent. (2分)

☐ T ☒ F

评测结果：答案正确 (2 分)

1-5 Mergesort is stable. (2分)

☒ T ☐ F

评测结果：答案正确 (2 分)

1-6 $(\log N)^3$ is $O(N)$. (2分)

☒ T ☐ F

评测结果：答案正确 (2 分)

1-7 Let P be the shortest path from S to T. If the weight of every edge in the graph is incremented by 1, P will still be the shortest path from S to T. (2分)

☐ T ☒ F

评测结果：答案正确 (2 分)

1-8 Quadratic probing is equivalent to double hashing with a secondary hash function of $Hash_2(k) = k$. (2分)

☒ T ☐ F

评测结果：答案错误 (0 分)

1-9 If keys are pushed onto a stack in the order `abcde`, then it's impossible to obtain the output sequence `cedab`. (2分)

☒ T ☐ F

评测结果：答案正确 (2 分)

1-10 If the inorder and the postorder traversal sequences of a binary tree have exactly the same order, then none of the nodes in the tree has a right subtree. (2分)

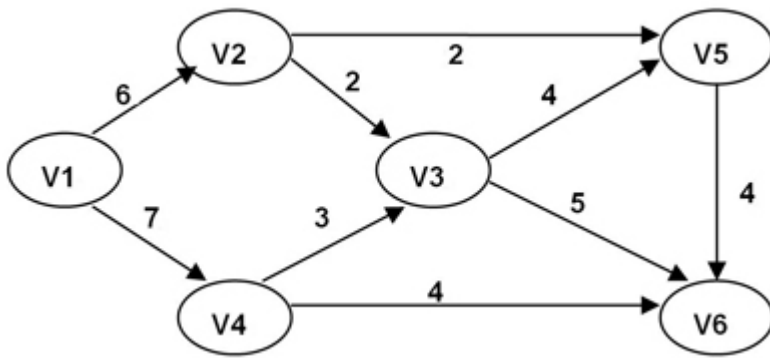
☒ T ☐ F

评测结果：答案正确 (2 分)

单选题

总分：60 得分：54

2-1 The maximum flow from v1 to v6 is __: (3分)



- ☒ A. 11
- ☐ B. 12
- ☐ C. 13
- ☐ D. 0

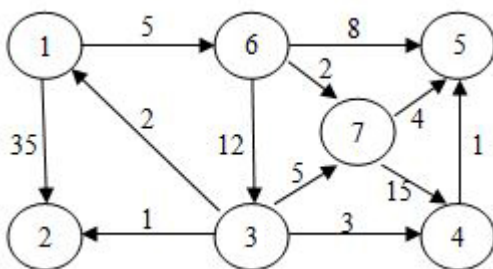
评测结果：答案正确 (3 分)

2-2 Given input {46, 79, 56, 38, 40, 84}. After the first partition (with the left most record as the pivot) of quick sort, the resulting sequence is: (3分)

- ☐ A. {38,46,79,56,40,84}
- ☐ B. {38,79,56,46,40,84}
- ☐ C. {38,46,56,79,40,84}
- ☒ D. {40,38,46,56,79,84}

评测结果：答案正确 (3 分)

2-3 Use Dijkstra algorithm to find the shortest paths from 1 to every other vertices. In which order that the destinations must be obtained? (3分)

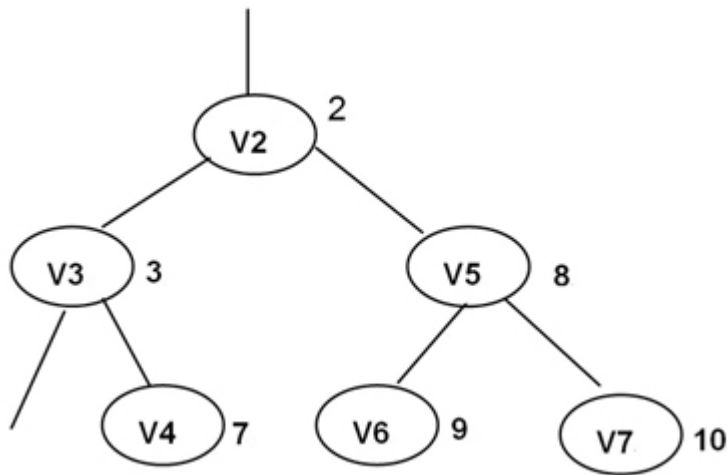


- ☒ A. 6, 7, 5, 3, 2, 4
- ☐ B. 6, 2, 5, 7, 3, 4
- ☐ C. 2, 3, 4, 5, 6, 7
- ☐ D. 2, 4, 3, 6, 5, 7

评测结果：答案正确 (3 分)

2-4 The following is the part of depth-first search tree to find the articulation points, and the Num(v)

value has been marked beside each vertex v . The back edges are not shown. Which of the following situation is impossible? (3分)



- ☐ A. $\text{low}(v3)$ is equaled to $\text{low}(v5)$
- ☒ B. $\text{low}(v5)$ is greater than $\text{low}(v7)$
- ☐ C. $\text{low}(v7)$ is 2
- ☐ D. $\text{low}(v4)$ is 1

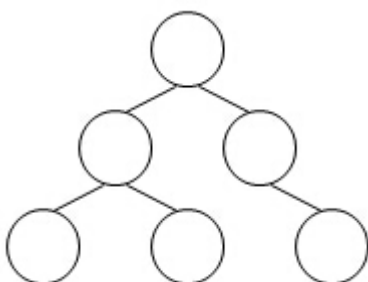
评测结果：答案正确 (3 分)

2-5 It is known that a 3-heap is a heap whose nodes have 3 children. Suppose that the level-order traversal sequence of a max-3-heap is {88, 76, 65, 82, 68, 46, 52, 44, 62, 33, 75, 28, 55, 60}. Use the linear algorithm to adjust this max-3-heap into a min-3-heap, and then run DeleteMin. As a result, there are __ nodes whose positions are not moved in the process. (3分)

- ☐ A. 2
- ☒ B. 3
- ☐ C. 4
- ☐ D. 5

评测结果：答案正确 (3 分)

2-6 Given the structure of a binary search tree (as shown in the figure), which one of the following insertion sequences is impossible? (3分)



- ☐ A. 85 56 89 95 18 75
- ☐ B. 85 56 75 89 18 95

- ☒ C. 85 89 75 56 18 95
- ☐ D. 85 89 95 56 75 18

评测结果：答案正确 (3 分)

2-7 Following is the C-like pseudo code of a function that takes a Queue as an argument.

```
void foo(Queue Q)
{
    Queue Q1 = CreateQueue(); // create an empty queue

    while (!IsEmpty(Q))
    {
        // dequeue an item from Q and enqueue it into Q1
        Enqueue(Q1, Dequeue(Q));
    }

    while (!IsEmpty(Q1))
    {
        // dequeue an item from Q1 and enqueue it into Q
        Enqueue(Q, Dequeue(Q1));
    }

    DisposeQueue(Q1);
}
```

What does the above function do? (3分)

- ☐ A. Removes the last item from Q
- ☒ B. Keeps Q unchanged
- ☐ C. Makes Q empty
- ☐ D. Reverses Q

评测结果：答案正确 (3 分)

2-8 Given a tree of degree 6. Suppose that the numbers of nodes of degrees 1, 2, 3, 4, 5, 6 are 3, 5, 1, 2, 4, 3, respectively. Then the number of leaf nodes must be: (3分)

- ☐ A. 35
- ☐ B. 39
- ☐ C. 43
- ☒ D. 45

评测结果：答案正确 (3 分)

2-9 Let T be a tree of N nodes created by union-by-size without path compression, then the minimum depth of T may be (3分)

- ☒ A. 1
- ☐ B. $\log N$
- ☐ C. $N - 1$
- ☐ D. $N/2$

评测结果：答案正确 (3 分)

2-10 In order to convert the infix expression `4 + 3 * (6 * 3 - 12)` to postfix expression using a stack S , then the minimum size of S must be: (3分)

- ☐ A. 2
- ☒ B. 3
- ☐ C. 4
- ☐ D. 5

评测结果：答案错误 (0 分)

2-11 A graph with 50 vertices and 17 edges must have at most __ connected component(s). (3分)

- ☐ A. 32
- ☐ B. 33
- ☒ C. 44
- ☐ D. 45

评测结果：答案正确 (3 分)

2-12 The inorder and the postorder traversal sequences of a binary tree are `a b c d e f g` and `a c b g f e d`, respectively. Then the preorder traversal sequences is: (3分)

- ☐ A. `d b a c f e g`
- ☒ B. `d b a c e f g`
- ☐ C. `d a c b f e g`
- ☐ D. `d c a b e f g`

评测结果：答案正确 (3 分)

2-13 Given input {15, 9, 7, 8, 20, -1, 4}. If the result of the 1st run of Shell sort is {15, -1, 4, 8, 20, 9, 7}, then the initial increment must be: (3分)

- ☐ A. 1
- ☐ B. 2

- ☐ C. 3
☒ D. 4

评测结果：答案正确 (3 分)

2-14 Suppose that the size of a hash table is 11, and the hash function is $H(\text{key}) = \text{key} \% 11$. The following 4 elements have been inserted into the table as $\text{Addr}(14)=3$, $\text{Addr}(38)=5$, $\text{Addr}(61)=6$, $\text{Addr}(86)=9$. When open addressing with quadratic probing is used to solve collisions, the address of the element with $\text{key}=49$ will be . (3分)

- ☐ A. 5
☒ B. 10
☐ C. 7
☐ D. 8

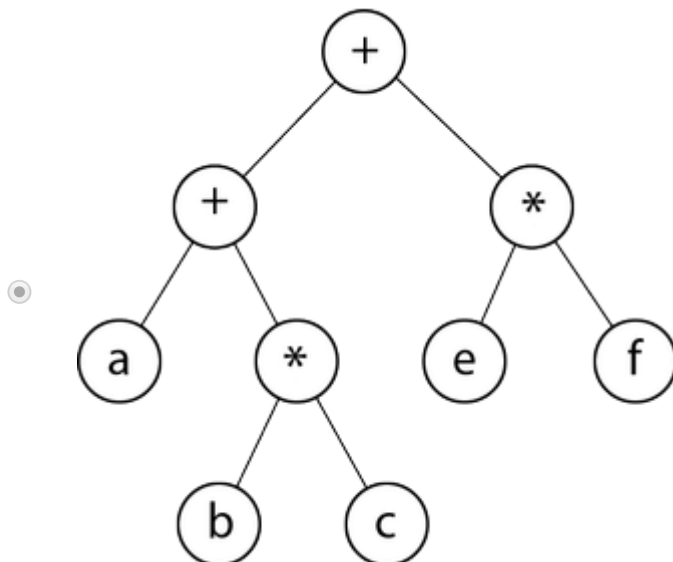
评测结果：答案正确 (3 分)

2-15 When inserting a new key **K** into a binary search tree **T** with 511 nodes, the worst-case number of comparisons between **K** and the keys already in **T** is in the range of: (3分)

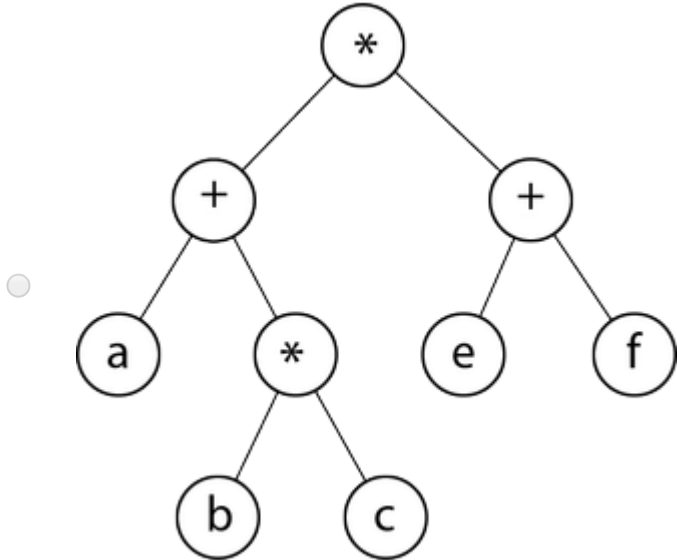
- ☐ A. [10, 511]
☒ B. [9, 511]
☐ C. [9, 512]
☐ D. [10, 512]

评测结果：答案正确 (3 分)

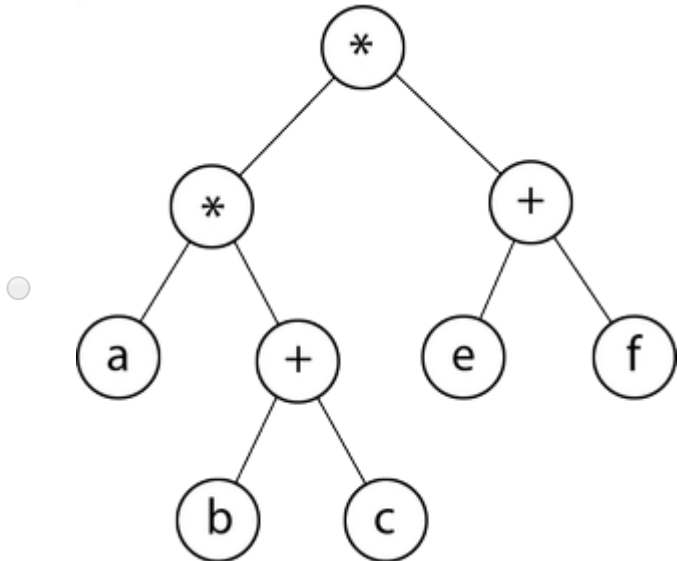
2-16 Which one of the following is the expression tree corresponding to the postfix expression **abc*+ef*+**? (3分)



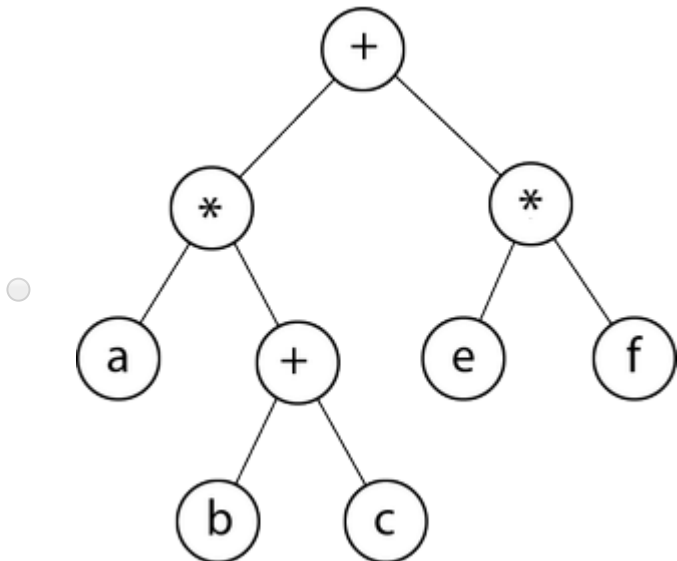
A.



B.



C.



D.

评测结果：答案正确 (3 分)

2-17 For an in-order threaded binary tree, if the pre-order and in-order traversal sequences are D A B C F

E and B A C D E F respectively, which pair of nodes' right links are both threads? (3分)

- ☐ A. D and A
- ☐ B. A and F
- ☐ C. B and A
- ☒ D. B and E

评测结果：答案正确 (3 分)

2-18 Which one of the following is a possible postorder traversal sequence of a binary search tree? (3分)

- ☐ A. 2 4 1 5 3 7 9 10 8 6
- ☐ B. 2 4 1 5 3 7 10 9 8 6
- ☒ C. 2 1 4 5 3 7 10 9 8 6
- ☐ D. 2 1 4 5 3 10 7 9 8 6

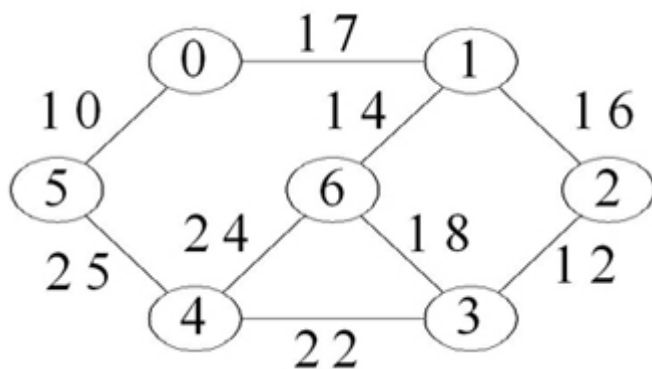
评测结果：答案正确 (3 分)

2-19 Given an initially empty hash table HT with length 7, together with a hash function $H(k) = k \% 7$. Let us use linear probing to solve collisions. What is the average search length for successful searches after inserting 22, 43, 15 one by one into HT? (3分)

- ☒ A. 1.5
- ☐ B. 1.6
- ☐ C. 2
- ☐ D. 3

评测结果：答案错误 (0 分)

2-20 To find the minimum spanning tree with Prim's algorithm for the following graph, a sequence of vertexes 6, 1, 2, 3 was found during the algorithm's early steps. Which one vertex will be added in the next step? (3分)



- ☒ A. 0
- ☐ B. 4
- ☐ C. 5

☐ D. the vertex serial is incorrect

评测结果：答案正确 (3 分)

程序填空题

总分：12 得分：9

5-1 The function is to find the -th smallest element in a list of elements. The function `BuildMaxHeap(H, K)` is to arrange elements ... into a max-heap. Please complete the following program.

```
ElementType FindKthSmallest ( int A[], int N, int K )
{
    /* it is assumed that K<=N */
    ElementType *H;
    int i, next, child;

    H = (ElementType *)malloc((K+1)*sizeof(ElementType));
    for ( i=1; i<=K; i++ ) H[i] = A[i-1];
    BuildMaxHeap(H, K);

    for ( next=K; next<N; next++ ) {
        H[0] = A[next];
        if ( H[0] < H[1] ) {
            for ( i=1; i*2<=K; i=child ) {
                child = i*2;
                if ( child!=K &&  (3分) ) child++;
                if (  (3分) )
                    H[i] = H[child];
                else break;
            }
            H[i] = H[0];
        }
    }
    return H[1];
}
```

评测结果：部分正确 (3 分)

序号	结果	得分
0	答案正确	3
1	答案错误	0

5-2 Given an array of integers, the function `MissingMin` is to find and return the minimum

positive integer which is **NOT** in the array. For example, given { 3, -1, 8, 1, 0 }, 2 is the smallest positive integer which is missing.

```
int MissingMin( int a[], int n )
{
    int i, j, min, missing=1;

    for( i=0; i<n; i++ ){
        min = i;
        for( j = i+1; j < n; j++ )
            if ( a[j]<a[min] (3分)) min = j;
        if ( min != i ) swap(a[i], a[min]);
        if ( a[i] == missing ) missing++;
        else if ( a[i] > missing ) break (3分);
    }
    return missing;
}
```

评测结果：答案正确 (6 分)

序号	结果	得分
0	答案正确	3
1	答案正确	3

函数题

总分：8 得分：8

6-1 Is Topological Order

Write a program to test if a give sequence **Seq** is a topological order of a given graph **Graph**.

Format of functions:

```
bool IsTopSeq( LGraph Graph, Vertex Seq[] );
```

where **LGraph** is defined as the following:

```
typedef struct AdjVNode *PtrToAdjVNode;
struct AdjVNode{
    Vertex AdjV;
    PtrToAdjVNode Next;
};

typedef struct Vnode{
    PtrToAdjVNode FirstEdge;
} AdjList[MaxVertexNum];
```

```
typedef struct GNode *PtrToGNode;
struct GNode{
    int Nv;
    int Ne;
    AdjList G;
};
typedef PtrToGNode LGraph;
```

The function `IsTopSeq` must return `true` if `Seq` does correspond to a topological order; otherwise return `false`.

Note: Although the vertices are numbered from 1 to MaxVertexNum, they are **indexed from 0** in the LGraph structure.

Sample program of judge:

```
#include <stdio.h>
#include <stdlib.h>

typedef enum {false, true} bool;
#define MaxVertexNum 10 /* maximum number of vertices */
typedef int Vertex;      /* vertices are numbered from 1 to MaxVertexNum */

typedef struct AdjVNode *PtrToAdjVNode;
struct AdjVNode{
    Vertex AdjV;
    PtrToAdjVNode Next;
};

typedef struct Vnode{
    PtrToAdjVNode FirstEdge;
} AdjList[MaxVertexNum];

typedef struct GNode *PtrToGNode;
struct GNode{
    int Nv;
    int Ne;
    AdjList G;
};
typedef PtrToGNode LGraph;

LGraph ReadG(); /* details omitted */

bool IsTopSeq( LGraph Graph, Vertex Seq[] );

int main()
{
    int i, j, N;
    Vertex Seq[MaxVertexNum];
```

```

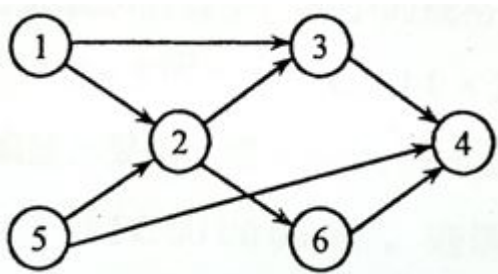
LGraph G = ReadG();
scanf("%d", &N);
for (i=0; i<N; i++) {
    for (j=0; j<G->Nv; j++)
        scanf("%d", &Seq[j]);
    if ( IsTopSeq(G, Seq)==true ) printf("yes\n");
    else printf("no\n");
}

return 0;
}

/* Your function will be put here */

```

Sample Input (for the graph shown in the figure):



```

6 8
1 2
1 3
5 2
5 4
2 3
2 6
3 4
6 4
5
1 5 2 3 6 4
5 1 2 6 3 4
5 1 2 3 6 4
5 2 1 6 3 4
1 2 3 4 5 6

```

Sample Output:

```

yes
yes
yes
no
no

```

代码

```
bool IsTopSeq(LGraph Graph, Vertex Seq[])
{
    int Indegree[MaxVertexNum] = {0};
    for (int i = 0; i < Graph->Nv; ++i)
    {
        if (Graph->G[i].FirstEdge)
        {
            PtrToAdjVNode p = Graph->G[i].FirstEdge;
            while (p)
            {
                Indegree[p->AdjV]++;
                p = p->Next;
            }
        }
    }

    for (int i = 0; i < Graph->Nv; ++i)
    {
        if (Indegree[Seq[i] - 1] == 0)
        {
            int v = Seq[i] - 1;
            PtrToAdjVNode p = Graph->G[v].FirstEdge;
            while (p)
            {
                Indegree[p->AdjV]--;
                p = p->Next;
            }
        }
        else
            return false;
    }
    return true;
}
```

评测结果：答案正确 (8 分)

测试点	结果	得分	耗时	内存
0	答案正确	4	2 ms	256 KB
1	答案正确	1	2 ms	384 KB
2	答案正确	1	2 ms	256 KB
3	答案正确	1	2 ms	256 KB
4	答案正确	1	28 ms	512 KB

a.c: In function 'ReadG':

a.c:68:2: warning: ignoring return value of 'scanf', declared with attribute warn_unused_re

```
scanf("%d", &Nv);    /* 读入顶点个数 */
```

```
^~~~~~
```

```
a.c:71:2: warning: ignoring return value of 'scanf', declared with attribute warn_unused_re
scanf("%d", &(Graph->Ne));    /* 读入边数 */
```

```
^~~~~~
```

```
a.c:76:7: warning: ignoring return value of 'scanf', declared with attribute warn_unused_re
scanf("%d %d", &E->V1, &E->V2);
```

```
^~~~~~
```

```
a.c: In function 'main':
```

```
a.c:90:5: warning: ignoring return value of 'scanf', declared with attribute warn_unused_re
scanf("%d", &N);
```

```
^~~~~~
```

```
a.c:93:13: warning: ignoring return value of 'scanf', declared with attribute warn_unused_r
scanf("%d", &Seq[j]);
```

```
^~~~~~
```

