

算法分析与设计第九周练习

开始时间 2022/04/15 22:00:00

结束时间 2022/04/20 23:59:00

答题时长 7319分钟

答卷类型 标准答案

总分 65

单选题

得分：暂无 总分：20

2-1 所求解的问题满足（ ）是使用动态规划算法的必要条件。 (5分)

- ☐ A. 子问题重叠
- ☐ B. 子问题互相独立
- ☐ C. 多米诺性质
- ☒ D. 最优子结构性质（优化原则）

2-2 We can tell that there must be a lot of redundant calculations during the exhaustive search for the matrix multiplication problem, because the search work load is the Catalan number, yet there are only ___ different sub-problems M_{ij} . (5分)

- ☐ A. $O(N)$
- ☒ B. $O(N^2)$
- ☐ C. $O(N^3)$
- ☐ D. $O(N^4)$

2-3 Why doesn't Floyd algorithm work if there are negative-cost cycles? (5分)

- ☐ A. Because Floyd didn't like negative numbers.
- ☒ B. Because Floyd algorithm will terminate after finite steps, yet the shortest distance is negative infinity if there is a negative-cost cycle.
- ☐ C. Because Floyd algorithm will fall into infinite loops.
- ☐ D. Because a negative-cost cycle will result in a negative $D[i][i]$, yet Floyd algorithm can only accept positive weights.

2-4 When solving the problem All-Pairs Shortest Path by Floyd method, which one of the following iterations can give us the correct answer? (5分)

- ☐ 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[ i ][ k ] + D[ k ][ j ] < D[ i ][ j ] )
                D[ i ][ j ] = D[ i ][ k ] + D[ k ][ j ];
```

```
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 ];
```

函数题

得分：暂无 总分：20

6-1 求解数字和为sum的方法数问题(动态规划法) (20分)

求解数字和为sum的方法数问题。给定一个有n个正整数的数组a和一个整数sum，求选择数组a中部分数字和为sum的方案数。若两种选取方案有一个数字的下标不一样，则认为是不同的方案。

函数接口定义：

```
long solve();
```

裁判测试程序样例：

```
#define MAXN 1001
#define MAXS 1001

using namespace std;
int a[MAXN];
long dp[MAXN][MAXS];
int n,sum;
long solve();

int main()
{
    cin>>n>>sum;
    for(int i=1;i<=n;i++)
        cin>>a[i];
    cout<<solve();
    return 0;
}

/* 请在这里填写答案 */
```

输入格式：

第1行为两个正整数n($1 \leq n \leq 1000$), sum($1 \leq \text{sum} \leq 1000$), 第2行为n个正整数 a_i , 以空格隔开。

```
5 15
5 5 10 2 3
```

输出格式：

输出方案数。

```
4
```

编程题

得分：暂无 总分：25

7-1 月饼 (25分)

月饼是中國人在中秋佳節時吃的一種傳統食品，不同地區有許多不同風味的月餅。現給定所有種類月餅的庫存量、總售價、以及市場的最大需求量，請你計算可以獲得的最大收益是多少。

注意：销售时允许取出一部分库存。样例给出的情形是这样的：假如我们有 3 种月饼，其库存量分别为 18、15、10 万吨，总售价分别为 75、72、45 亿元。如果市场的最大需求量只有 20 万吨，那么我们最大收益策略应该是卖出全部 15 万吨第 2 种月饼、以及 5 万吨第 3 种月饼，获得 $72 + 45/2 = 94.5$ （亿元）。

输入格式：

每个输入包含一个测试用例。每个测试用例先给出一个不超过 1000 的正整数 N 表示月饼的种类数、以及不超过 500（以万吨为单位）的正整数 D 表示市场最大需求量。随后一行给出 N 个正数表示每种月饼的库存量（以万吨为单位）；最后一行给出 N 个正数表示每种月饼的总售价（以亿元为单位）。数字间以空格分隔。

输出格式：

对每组测试用例，在一行中输出最大收益，以亿元为单位并精确到小数点后 2 位。

输入样例：

```
3 20
18 15 10
75 72 45
```

输出样例：

```
94.50
```
