

西北工业大学研究生院

学位研究生课程考试试题

考试科目：算法设计与分析

课程编号：M10M21002

考核形式：考查

考试时间：

开课学期：2019-2020 春季

任课教师：高翔

说明：有答案必须写在答题册上, 否则无效。

共 1 页 第 1 页

学院：计算机学院

姓名：郭炜

学号：2019202243

1. 设有 n 个顾客同时等待一项服务, 顾客 i 需要的时间为 t_i , $i = 1, 2, \dots, n$. 从时刻0开始计时, 若在时刻 t 开始对顾客 i 服务, 那么 i 的等待时间就是 t , 应该怎样安排 n 个顾客的服务次序使得总的等待时间(每个顾客等待时间的总和)最少? 假设多个顾客服务时间分别为 $\{1, 3, 2, 15, 10, 6, 12\}$, 请求出该问题的解。

1) 解题思路

题目要求总等待时间最小, 将每个顾客需要的服务时间从小到大排序。

如此每个顾客要等待的时间为 $t_1 = t_1$, $t_2 = t_1 + t_2$, $t_3 = t_1 + t_2 + t_3 \cdots t_n = t_1 + t_2 + \cdots t_n$.

总等待时间为 $t = t_1 \times n + t_2 \times (n - 1) + \cdots t_n$ 。

2) 代码

<https://github.com/SnailForce/The-Design-And-Analysis-Of->

[Algorithms/blob/master/Assessment/Project1/Project1/Source.cpp](https://github.com/SnailForce/The-Design-And-Analysis-Of-Algorithms/blob/master/Assessment/Project1/Project1/Source.cpp)

```
#include <iostream>
#include <vector>
#include <fstream>
#include <algorithm>
#include <numeric>
using namespace std;

unsigned int n = 0;
vector<int> result;
vector<vector<int>> u;

int main()
{
```

西北工业大学研究生院

学位研究生课程考试试题

考试科目：算法设计与分析

考核形式：考查

开课学期：2019-2020 春季

说明：有答案必须写在答题册上, 否则无效。

课程编号：M10M21002

考试时间：

任课教师：高翔

共 1 页 第 1 页

```
ifstream is("./data.txt", ifstream::in);
ofstream os("./res.txt", ofstream::out);

is >> n;
for (size_t i = 0; i < n; i++)
{
    int cur;
    is >> cur;
    u.push_back({cur, int(i) + 1});
}
sort(u.begin(), u.end(), [](const vector<int>& u1, const vector<int>& u2) {return u1.at(0) < u2.
at(0); });
for (size_t i = 0; i < n; i++)
{
    int t = 0;
    for (size_t j = 0; j < i; j++)
    {
        t += u.at(j).at(0);
    }
    result.push_back(t);
}
int sum = 0;
for (auto i = result.begin(); i < result.end(); i++)
{
    sum += i.operator[](0);
}
os << "sorted data: " << endl;
for (size_t i = 0; i < n; i++)
{
    os << u.at(i).at(0) << " " << u.at(i).at(1) << endl;
}
os << endl << "result: " << endl;
for (size_t i = 0; i < n; i++)
{
    os << result.at(i) << " ";
}
os << endl;
os << "sum: " << sum << endl;
```

西北工业大学研究生院

学位研究生课程考试试题

考试科目：算法设计与分析

考核形式：考查

开课学期：2019-2020 春季

说明：有答案必须写在答题册上, 否则无效。

课程编号：M10M21002

考试时间：

任课教师：高翔

共 1 页 第 1 页

```
return 0;  
}
```

Data.txt



```
data.txt - 记事本  
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)  
7  
1 3 2 15 10 6 12  
第 2 行, 第 17 列 100% Windows (CRLF) UTF-8
```

3) 程序运行结果截图

Res.txt

Sorted data 的第一列是等待时间的从小到大的排序, 第二列是每位顾客的编号 (从 1 到 7)。

Result 是每位顾客需要等待的时间

Sum 是总等待时间



```
res.txt - 记事本  
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)  
sorted data:  
1 1  
2 3  
3 2  
6 6  
10 5  
12 7  
15 4  
  
result:  
0 1 3 6 12 22 34  
sum: 78  
第 1 行, 第 1 列 100% Windows (CRLF) UTF-8
```

西北工业大学研究生院

学位研究生课程考试试题

考试科目：算法设计与分析

课程编号：M10M21002

考核形式：考查

考试时间：

开课学期：2019-2020 春季

任课教师：高翔

说明：有答案必须写在答题册上, 否则无效。

共 1 页 第 1 页

2. 设有 n 项任务由 k 个可并行操作的机器完成, 完成任务 i 所需时间是 t_i , 求一个最佳任务分配方案, 使得完成时间(从0开始计时, 到最后一台机器终止)达到最短。假设有10个任务所需时间为{7, 9, 5, 1, 3, 2, 15, 10, 6, 12}, 有3台机器可用, 请求出问题的解。

1) 解题思路

分支限界法

若当前已完成任务的时间总和大于历史最佳时间, 则剪枝回溯, 无须进行后续任务的遍历;

若当前机器已用时间加上当前任务所需时间大于历史最佳时间, 则越过该机器;

2) 代码

<https://github.com/SnailForce/The-Design-And-Analysis-Of->

[Algorithms/blob/master/Assessment/Project2/Project2/Source.cpp](https://github.com/SnailForce/The-Design-And-Analysis-Of-Algorithms/blob/master/Assessment/Project2/Project2/Source.cpp)

```
#include <iostream>
#include <vector>
#include <fstream>
#include <algorithm>
using namespace std;
constexpr auto MAX = 20;
int cur = 0, result = 10000;
vector<int> data;
vector<int> machine;
vector<vector<int>> task;
vector<vector<int>> res;
void getData(ifstream& is)
{
    int tmp;
    // machine num
    is >> tmp;
```

西北工业大学研究生院

学位研究生课程考试试题

考试科目：算法设计与分析

考核形式：考查

开课学期：2019-2020 春季

说明：有答案必须写在答题册上, 否则无效。

课程编号：M10M21002

考试时间：

任课教师：高翔

共 1 页 第 1 页

```
::machine.assign(tmp, 0);
while (!is.eof())
{
    is >> tmp;
    ::data.push_back(tmp);
}
vector<int> t(::data.size(), 0);
for (size_t i = 0; i < ::machine.size(); i++)
{
    task.push_back(t);
}
}

void dfs(unsigned int u)
{
    if (u == ::data.size())
    {
        // max machine element
        cur = *max_element(machine.begin(), machine.end());
        // update result
        if (cur < result)
        {
            result = cur;
            ::res.assign(::task.begin(), ::task.end());
            return;
        }
    }
    else
    {
        for (size_t i = 0; i < ::machine.size(); i++)
        {
            if (::machine.at(i) + ::data.at(u) > result)
            {
                continue;
            }
            else
            {
                ::machine.at(i) += ::data.at(u);
```

西北工业大学研究生院

学位研究生课程考试试题

考试科目：算法设计与分析

考核形式：考查

开课学期：2019-2020 春季

说明：有答案必须写在答题册上, 否则无效。

课程编号：M10M21002

考试时间：

任课教师：高翔

共 1 页 第 1 页

```
        ::task.at(i).push_back(::data.at(u));
        dfs(u + 1);
        ::machine.at(i) -= ::data.at(u);
        ::task.at(i).pop_back();
    }
}
}
}
int main()
{
    ifstream is("./data.txt", ifstream::in);
    // input data
    getData(is);
    // reverse data
    sort(::data.rbegin(), ::data.rend());
    dfs(0);
    cout << "result: " << ::result << endl;
    int index = 1;
    for (auto m : ::res)
    {
        cout << "machine " << index << ": ";
        for (auto t : m)
        {
            if (t != 0)
            {
                cout << t << " ";
            }
        }
        cout << endl;
        ++index;
    }
}
```

西北工业大学研究生院

学位研究生课程考试试题

考试科目：算法设计与分析

考核形式：考查

开课学期：2019-2020 春季

说明：有答案必须写在答题册上, 否则无效。

课程编号：M10M21002

考试时间：

任课教师：高翔

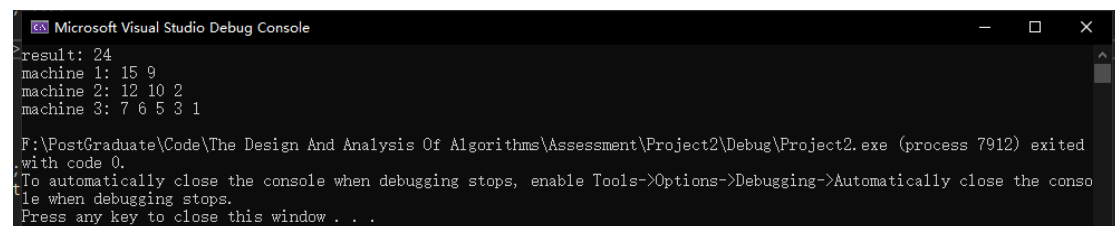
共 1 页 第 1 页

Data.txt



```
3
7 9 5 1 3 2 15 10 6 12
```

3) 程序运行结果截图



```
result: 24
machine 1: 15 9
machine 2: 12 10 2
machine 3: 7 6 5 3 1

F:\PostGraduate\Code\The Design And Analysis Of Algorithms\Assessment\Project2\Debug\Project2.exe (process 7912) exited
with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

本题结果并未像第一题一样，输出每个任务对应的索引值，如果需要可以按照第一题的形式增加，由于数据量较少且将机器和任务所需时间直接关联输出，较为容易理解。

西北工业大学研究生院

学位研究生课程考试试题

考试科目：算法设计与分析

考核形式：考查

开课学期：2019-2020 春季

说明：有答案必须写在答题册上, 否则无效。

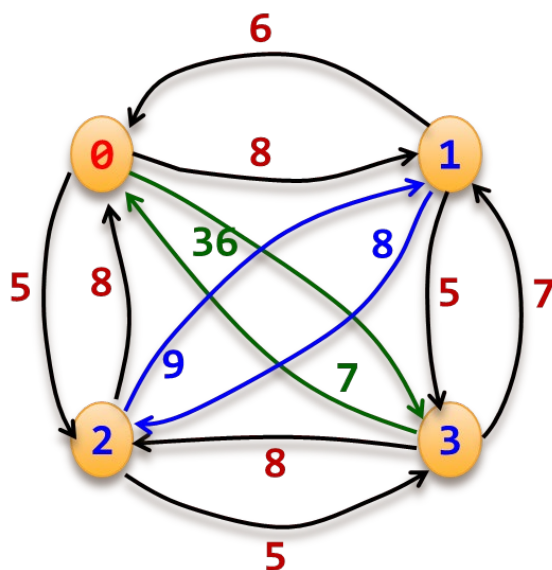
课程编号：M10M21002

考试时间：

任课教师：高翔

共 1 页 第 1 页

3. 假设有一个旅行商人从0号城市出发拜访4个城市，他选择所要走的路径，路径的限制是每个城市只能拜访一次，而且最后要回到原来出发的城市。路径的选择目标是要求得的路径路程为所有路径之中的最小值。



1) 解题思路

动态规划

假设从顶点 s 出发, 令 $d(i, V)$ 表示从顶点 i 出发经过 V (是一个点的集合)中各个顶点一次且仅一次, 最后回到出发点 s 的最短路径长度。

dp 方程为

$$d(i, V) = \begin{cases} c_{is}, & V = \phi, i \neq s \\ \min\{c_{ik} + d(k, V - \{K\})\}, & k \in V, V \neq \phi \end{cases}$$

由于数据量较小, 分支限界法也可以解决该问题, 在数据规模稍大一点, 动态规划优于分支限界法, 如果数据量非常大, 则两种方法都不适用, 蚁群算法或遗传算法求解 TSP 问题更为合适。

西北工业大学研究生院

学位研究生课程考试试题

考试科目：算法设计与分析

考核形式：考查

开课学期：2019-2020 春季

说明：有答案必须写在答题册上, 否则无效。

课程编号：M10M21002

考试时间：

任课教师：高翔

共 1 页 第 1 页

2) 代码

<https://github.com/SnailForce/The-Design-And-Analysis-Of->

[Algorithms/blob/master/Assessment/Project3/Project3/Source.cpp](https://github.com/SnailForce/The-Design-And-Analysis-Of-Algorithms/blob/master/Assessment/Project3/Project3/Source.cpp)

```
#include <iostream>
#include <vector>
#include <fstream>
#include <algorithm>
using namespace std;

constexpr auto N = 10;
constexpr auto INF = 10000;

class TSP
{
private:
    size_t numOfCity;
    size_t r;
    vector<vector<int>> distance;
    vector<vector<int>> dp;
public:
    TSP();
    void getShortestDistance();
    void printDP();
};

TSP::TSP()
{
    cout << "input citys' num " << endl;
    cin >> numOfCity;
    r = 1 << (numOfCity - 1); // dp 矩阵每一行的维度
    cout << "input citys' distance " << endl;
    /*
    10000 8 5 36
    6 10000 9 5
    */
}
```

西北工业大学研究生院

学位研究生课程考试试题

考试科目：算法设计与分析

考核形式：考查

开课学期：2019-2020 春季

说明：有答案必须写在答题册上，否则无效。

课程编号：M10M21002

考试时间：

任课教师：高翔

共 1 页 第 1 页

```
8 8 10000 5
7 7 8 10000
*/
for (size_t i = 0; i < numOfCity; i++)
{
    vector<int> tmp;
    for (size_t j = 0; j < numOfCity; j++)
    {
        int u;
        cin >> u;
        tmp.push_back(u);
    }
    distance.push_back(tmp);
}
for (size_t i = 0; i < numOfCity; i++)
{
    vector<int> tmp;
    tmp.assign(r, INF); // 初始化为最大值
    dp.push_back(tmp);
}
}

void TSP::getShortestDistance()
{
    for (size_t i = 0; i < numOfCity; i++)
    {
        dp.at(i).at(0) = distance.at(i).at(0); // 第一列相当于从源点直接到达目的点
    }
    for (size_t j = 1; j < r; j++)
    {
        for (size_t i = 0; i < numOfCity; i++)
        {
            // 判断该组合是否包含源点
            if (i > 0 && ((j >> (i - 1)) & 1) == 1)
            {
                continue;
            }
        }
    }
}
```

西北工业大学研究生院

学位研究生课程考试试题

考试科目：算法设计与分析

考核形式：考查

开课学期：2019-2020 春季

说明：有答案必须写在答题册上, 否则无效。

课程编号：M10M21002

考试时间：

任课教师：高翔

共 1 页 第 1 页

```
for (size_t k = 1; k < numOfCity; k++)
{
    // 判断 k 城市是否在当前组合 j 中
    if ((j >> (k - 1) & 1) == 1)
    {
        // dp 方程
        if (dp.at(i).at(j) > distance.at(i).at(k) + dp.at(k).at(j ^ (1 << (k - 1))))
        {
            dp.at(i).at(j) = distance.at(i).at(k) + dp.at(k).at(j ^ (1 << (k - 1)));
        }
    }
}

// 打印 dp 数组
void TSP::printDP()
{
    for (size_t i = 0; i < numOfCity; i++)
    {
        for (size_t j = 0; j < r; j++)
        {
            cout << dp.at(i).at(j) << "\t";
        }
        cout << endl;
    }

    // 第一行最后一个刚好是结果
    cout << "The length of the shortest path is " << dp.at(0).back() << endl;
}

int main()
{
    auto task = new TSP();
    task->getShortestDistance();
    task->printDP();
    return 0;
}
```

西北工业大学研究生院

学位研究生课程考试试题

考试科目：算法设计与分析

考核形式：考查

开课学期：2019-2020 春季

说明：有答案必须写在答题册上, 否则无效。

课程编号：M10M21002

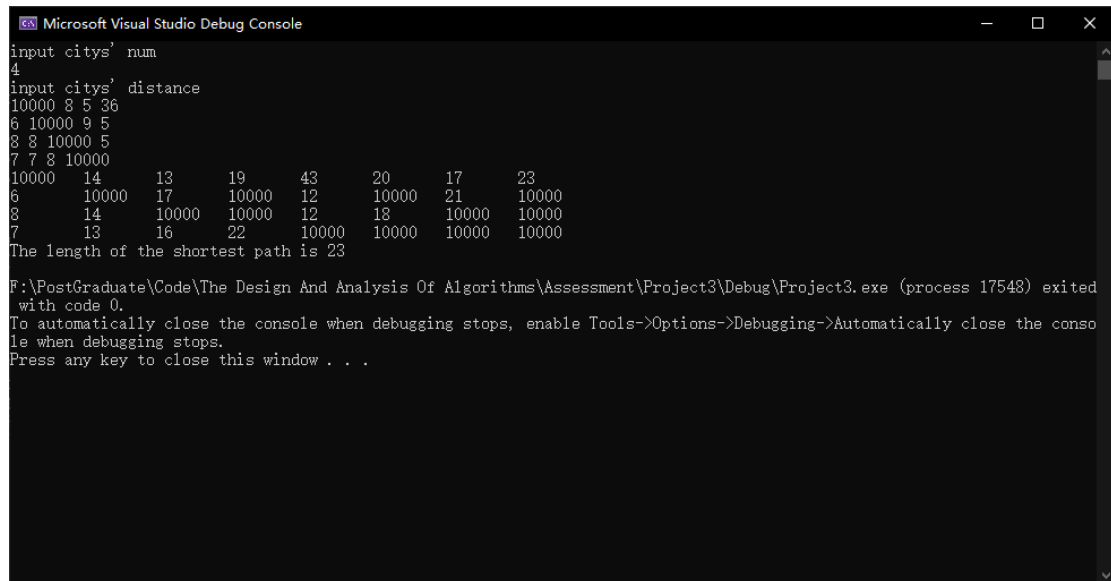
考试时间：

任课教师：高翔

共 1 页 第 1 页

3) 程序运行结果截图

10000 设置的是一个极大值



```
Microsoft Visual Studio Debug Console
input citys' num
4
input citys' distance
10000 8 5 36
6 10000 9 5
8 8 10000 5
7 7 8 10000
10000 14 13 19 43 20 17 23
6 10000 17 10000 12 10000 21 10000
8 14 10000 10000 12 18 10000 10000
7 13 16 22 10000 10000 10000 10000
The length of the shortest path is 23
F:\PostGraduate\Code\The Design And Analysis Of Algorithms\Assessment\Project3\Debug\Project3.exe (process 17548) exited
with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console
when debugging stops.
Press any key to close this window . . .
```