

**实验（实习）报告**

实验(实习)名称：实验四

日期：2022.5.26

学院：应用技术学院

专业：计算机科学与技术

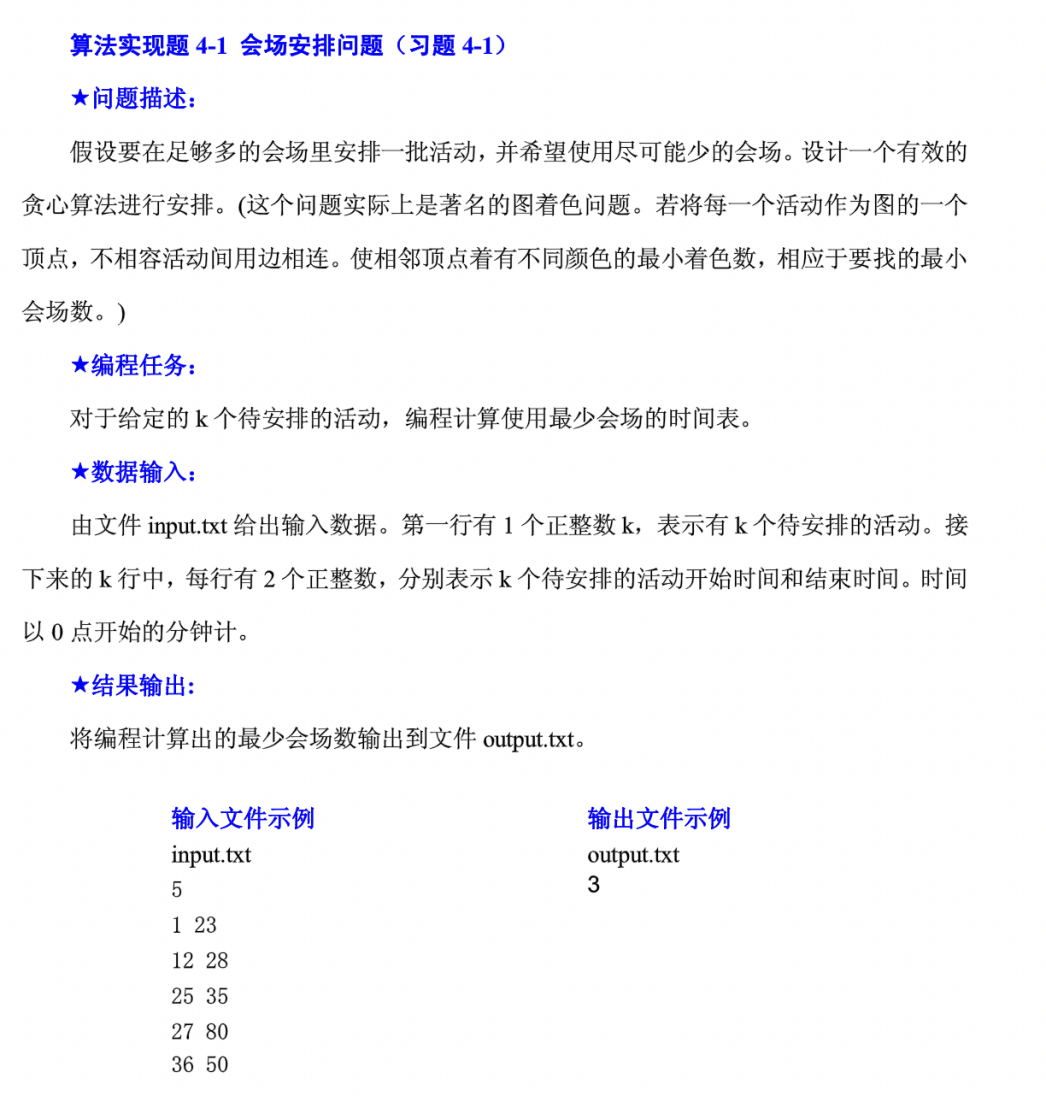
班级：1班

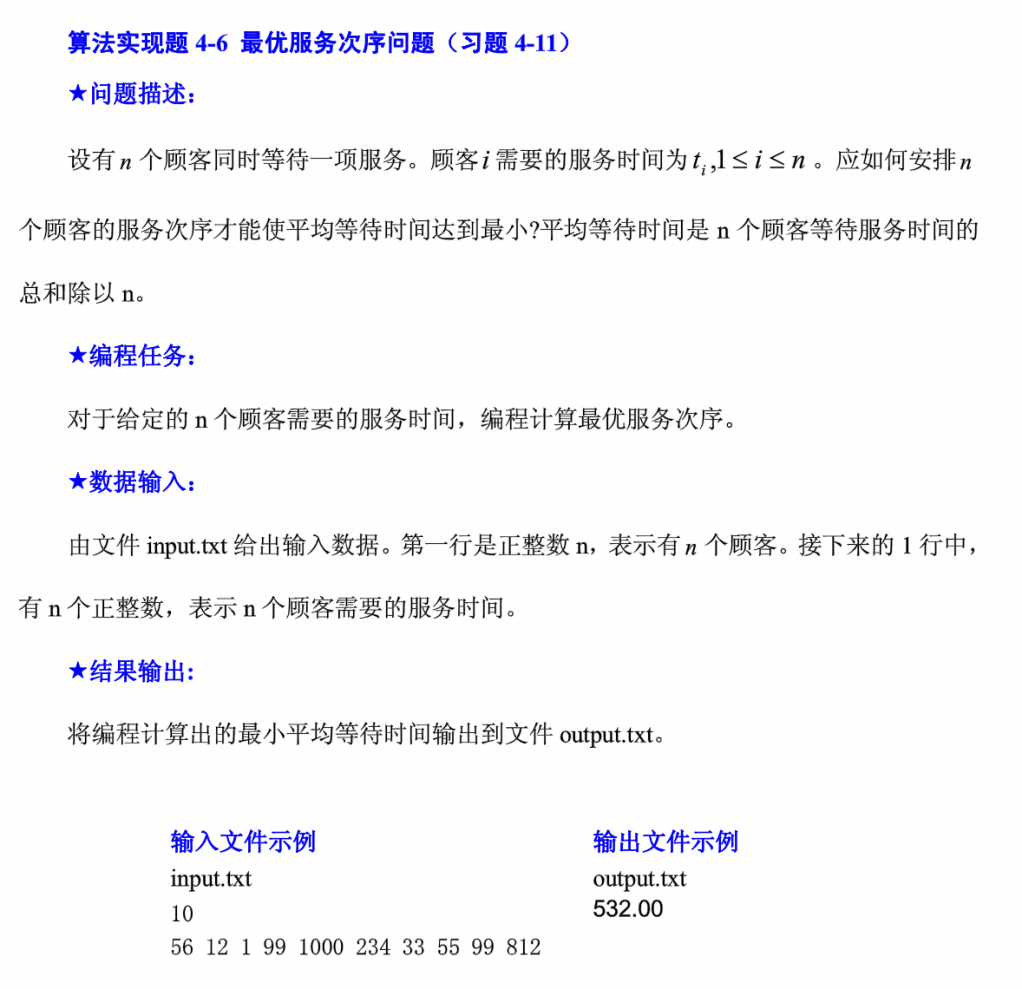
姓名：成凯

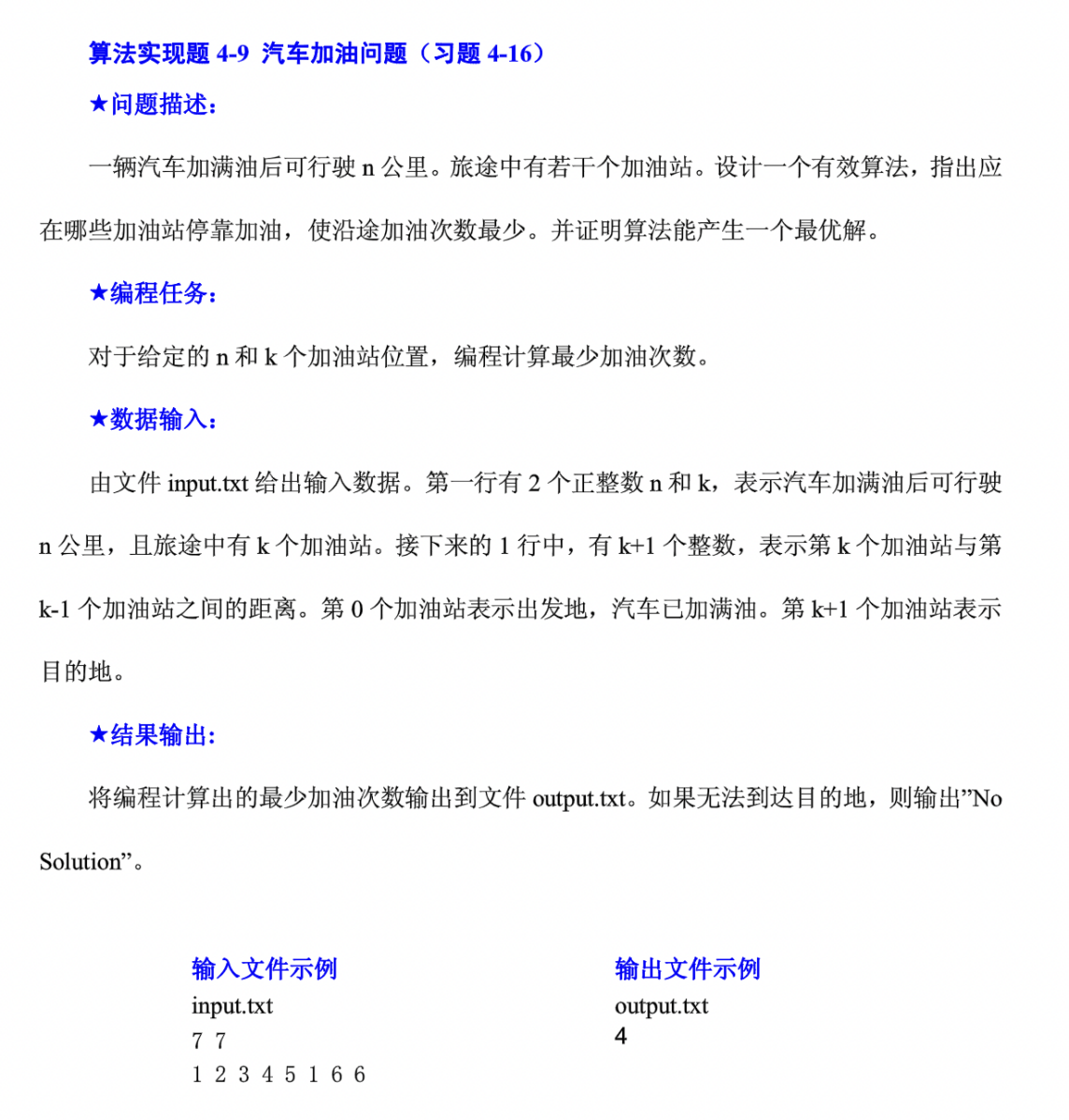
学号：201833050025

指导教师：庞亚伟

1. 实验目的
2. 实验内容







1. 实验步骤和实验结果

算法一实现代码：

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

ios::sync\_with\_stdio(false);

cin.tie(NULL);

cout.tie(NULL);

int n, i, Begins[10002], Ends[10002]; // Begins开始时间, Ends结束时间

cin >> n;

for (i = 0; i < n; i++)

cin >> Begins[i] >> Ends[i];

sort(Begins, Begins + n); // 升序排序

sort(Ends, Ends + n); // 升序排序

int j = 0, ans = 0;

for (i = 0; i < n; i++)

if (Begins[i] < Ends[j]) // 如果开始时间小于结束时间, 会场数加1

ans++;

else // 开始时间大于等于结束时间, 会场不用增加, 换下一家活动

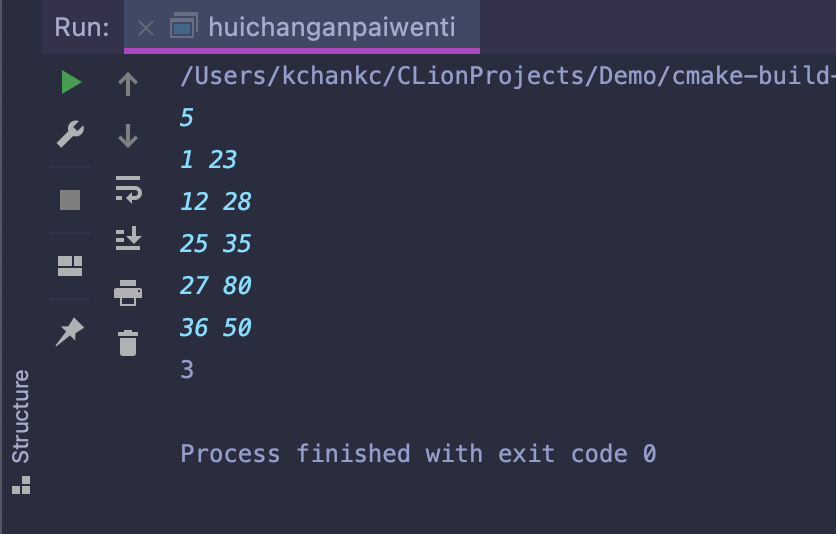
j++;

cout << ans << endl;

return 0;

}

算法一运行结果：



算法二实现代码：

#include<stdio.h>

#include<algorithm>

#define n 10

#define s 2

using namespace std;

double fun1(int\* list) { //最优服务次序问题

int i, sum = 0;

for (i = 0; i < n; i++)

sum += list[i] \* (n - i);

return sum / (n \* 1.0);

}

double fun2(int\* list) { //多次最优服务次序问题

int i, sum = 0;

int\* temp = new int[s];

for (i = 0; i < s; i++) {

temp[i] = 0;

}

for (i = 0; i < n; i++) {

temp[i % s] += list[i];

sum += temp[i % s];

}

return sum / (n \* 1.0);

}

int main() {

int i;

int list[n] = {56, 12, 1, 99, 1000, 234, 33, 55, 99, 812};

sort(list, list + n);

for (i = 0; i < n; i++) {

printf("%d\t", list[i]);

}

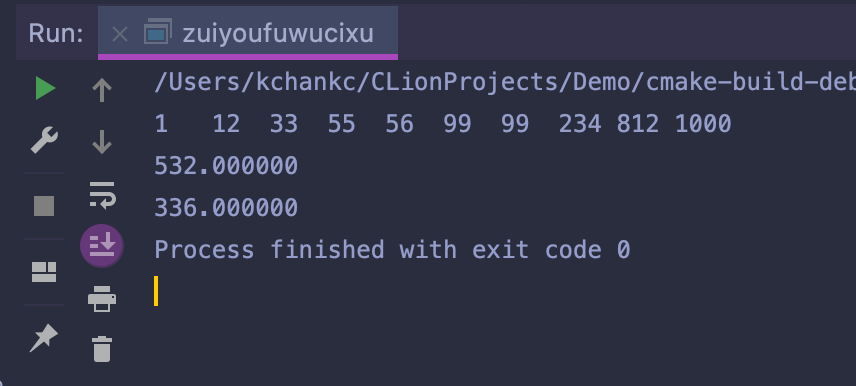
printf("\n%1f", fun1(list));

printf("\n%1f", fun2(list));

return 0;

}

算法二运行结果：



算法三实现代码：

#include<cstdio>

#include<iostream>

#include<algorithm>

#include<cstring>

#include<queue>

#define maxn 1000010

#define maxm 3000010

#define inf 1000000000

using namespace std;

int head[maxn], dis[maxn], n, k, a, b, c, cnt;

bool inq[maxn];

int map[110][110];

struct edge {

int next;

int to;

int val;

} e[maxm];

void insert(int u, int v, int w) {

e[++cnt].next = head[u];

head[u] = cnt;

e[cnt].to = v;

e[cnt].val = w;

}

queue<int> q;

void spfa(int s) {

memset(dis, 0x3f, sizeof(dis));

memset(inq, false, sizeof(inq));

dis[s] = 0;

inq[s] = true;

q.push(s);

while (!q.empty()) {

int now = q.front();

q.pop();

inq[now] = false;

for (int i = head[now]; i; i = e[i].next) {

int v = e[i].to;

if (dis[v] > dis[now] + e[i].val) {

dis[v] = dis[now] + e[i].val;

if (!inq[v]) {

q.push(v);

inq[v] = true;

}

}

}

}

}

int getid(int x, int y, int level) {

return ((x - 1) \* n + y) + n \* n \* (k - level);

}

int main() {

cin >> n >> k >> a >> b >> c;

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= n; j++) scanf("%d", &map[i][j]);

}

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= n; j++) {

if (map[i][j] == 1) {

for (int l = 0; l <= k - 1; l++) insert(getid(i, j, l), getid(i, j, k), a);

if (i < n) insert(getid(i, j, k), getid(i + 1, j, k - 1), 0);

if (j < n) insert(getid(i, j, k), getid(i, j + 1, k - 1), 0);

if (i > 1) insert(getid(i, j, k), getid(i - 1, j, k - 1), b);

if (j > 1) insert(getid(i, j, k), getid(i, j - 1, k - 1), b);

} else {

for (int l = 0; l <= k - 1; l++) insert(getid(i, j, l), getid(i, j, k), a + c);

for (int l = 1; l <= k; l++) {

if (i < n) insert(getid(i, j, l), getid(i + 1, j, l - 1), 0);

if (j < n) insert(getid(i, j, l), getid(i, j + 1, l - 1), 0);

if (i > 1) insert(getid(i, j, l), getid(i - 1, j, l - 1), b);

if (j > 1) insert(getid(i, j, l), getid(i, j - 1, l - 1), b);

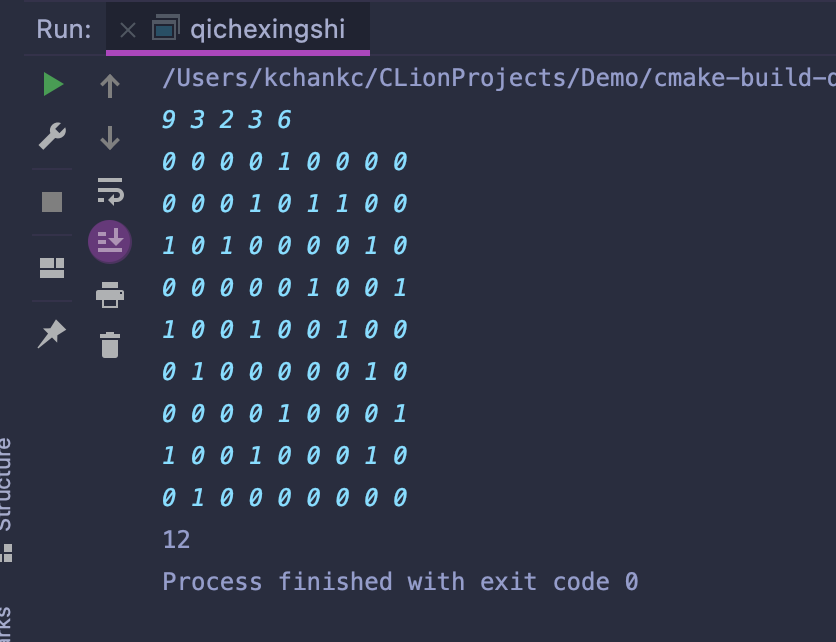
}

}

}

}

算法三运行结果：



1. 分析与讨论

通过本次实验再次加深了对动态规划算法的理解，总而言之透过本次实验让我受益匪浅。