《算法分析与设计》

课后作业

作业编号 作业2

学 号

姓 名

专 业 软件工程

学 院 计算机与人工智能学院

二0二二年三月

第2章 算法效率分析

1．阅读下面的程序，按要求回答下面的问题。

1. #include <stdio.h>
2. struct device{
3. int bw,price;
4. };
5. void merge(int \*a,int p,int q,int r,int \*b)
6. {
7. int i,j,k;
8. i=p;
9. j=q+1;
10. k=p;
11. for(;;){
12. if(i>q){
13. for(i=j;i<=r;i++)
14. b[k++]=a[i];
15. break;
16. }
17. if(j>r){
18. for(j=i;j<=q;j++)
19. b[k++]=a[j];
20. break;
21. }
22. if(a[i]<=a[j])
23. b[k++]=a[i++];
24. else
25. b[k++]=a[j++];
26. }
27. for(k=p;k<=r;k++)
28. a[k]=b[k];
29. }
30. void merge\_sort(int \*a,int p,int r,int \*b)
31. {
32. int q;
33. if(p<r){
34. q=(p+r)/2;
35. merge\_sort(a,p,q,b);
36. merge\_sort(a,q+1,r,b);
37. merge(a,p,q,r,b);
38. }
39. }
40. int main()
41. {
42. int c,n,i,j,k,s,t,q,d[101],band[10000],temp[10000],minw,maxw[101],minp;
43. long totalp;
44. float maxbp;
45. struct device a[101][101];
46. scanf("%d",&c);
47. for(i=1;i<=c;i++){
48. scanf("%d",&n);
49. q=1;
50. for(j=1;j<=n;j++){
51. scanf("%d",&d[j]);
52. maxw[j]=0;
53. for(k=1;k<=d[j];k++){
54. scanf("%d %d",&a[j][k].bw,&a[j][k].price);
55. band[q++]=a[j][k].bw;
56. if(a[j][k].bw>maxw[j])
57. maxw[j]=a[j][k].bw;
58. }
59. }
60. minw=32767;
61. for(j=1;j<=n;j++)
62. if(maxw[j]<minw)
63. minw=maxw[j];
64. maxw[0]=minw;
65. merge\_sort(band,1,q-1,temp);
66. maxbp=0;
67. band[0]=-1;
68. for(j=1;band[j]<=maxw[0];j++){
69. if(band[j]!=band[j-1]){
70. minw=band[j];
71. totalp=0;
72. for(s=1;s<=n;s++){
73. minp=32767;
74. for(t=1;t<=d[s];t++){
75. if(a[s][t].bw>=minw&&a[s][t].price<minp)
76. minp=a[s][t].price;
77. }
78. totalp+=minp;
79. }
80. if(minw\*1.0/totalp>maxbp)
81. maxbp=minw\*1.0/totalp;
82. }
83. }
84. printf("%.3f\n",maxbp);
85. }
86. return 0;
87. }

样例输入：

1 5

3 100 30 150 40 120 50

2 80 40 100 50

4 120 80 100 40 150 45 160 50

2 140 80 100 40

3 150 40 200 80 180 60

要求：

1. 写出在样例输入时，程序执行到66行时q的值以及数组元素band[0]到band[q-1]的值。（10分，每错一个扣1分，直到扣完为止）

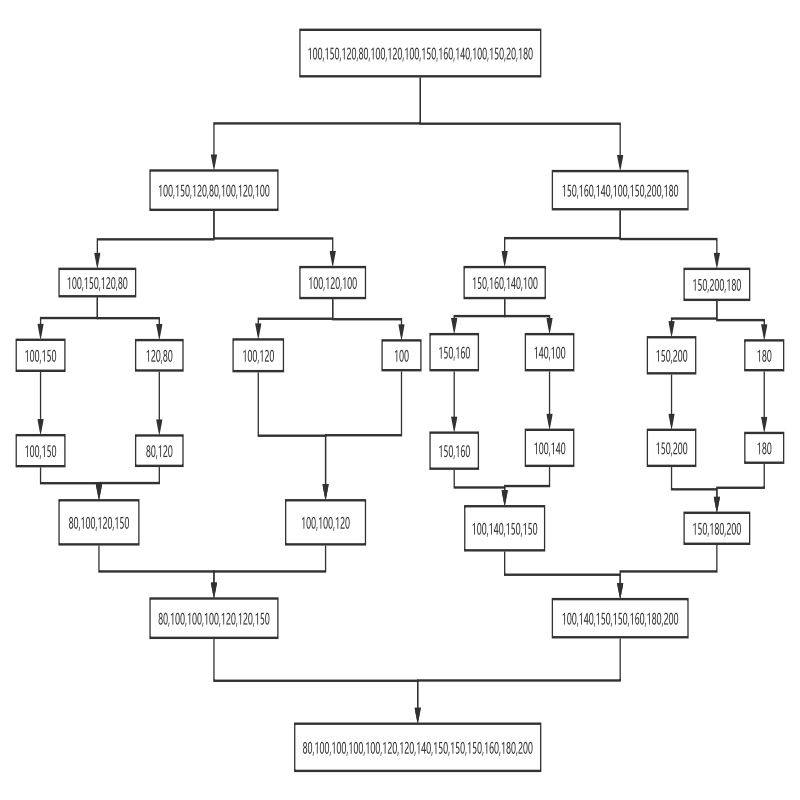
q = 15,

band:[0] = 0,

1. = 80,
2. = 100,
3. = 100,
4. = 100,
5. = 100,
6. = 120,
7. = 120,
8. = 140,
9. = 150,
10. = 150,
11. = 150,
12. = 160,
13. = 180,
14. = 200
15. 在样例输入时，分析函数merge\_sort的执行过程。 (30分)

该函数就是运用分治的思想，把排序的过程变为先把数组分成左右两个部分，分别排序，再将排好序的两个数组合并成一个有序数组。

树形结构如图所示：



1. 分析函数merge的时间复杂度，写出分析过程。（15分）

对于第一个for循环，时间复杂度取决于内部运行结果，内嵌for循环均为单层循环，时间复杂度为O(n），因此第一个for循环的时间复杂度为O（n）；

对于第二个for循环，循环次数为r-p，时间复杂度为O(n);

因此merge函数的时间复杂度为O(n).

1. 分析函数merge\_sort的时间复杂度，写出分析过程。（15分）
2. 假设处理的数据规模大小为 N，运行时间设为：T(N)  
   ① 当把 N 分为两半时，那么处理大小为 N/2 子数组花费时间为：T(N/2)  
   ② 合并花费时间与数据规模成正比:N  
   所以处理规模大小为N的数据所需要花费两个大小为 N/2 的子数组加上合并花费的时间,即：

T(N) = 2T(N/2) + N;

T(N)/N=T(N/2)/(N/2)+1;

令N=N/2得：

T(N/2)/(N/2)=T(N/4)/(N/4)+1;

递归可得：

T(2)/2=T(1)/1+1;

将上述方程相加得：

T(N)/N=T(1)/1+logN;

又

T(1)=1;

最终求得时间复杂度为O( nlogn )

1. 写出在样例输入时，程序第2次执行76行对应的语句之后变量minp的值，并写出分析过程。（20分）

minp = 40

此时maxw[1]=150,maxw[2]=100,maxw[3]=160,maxw[4]=140,maxw[5]=200,

其中最小值minw=100；band[1]=80，进入第68行for循环；

此时j=1，band[1]=80<maxw[0]=100，满足条件，执行判断，band[1]！=band[0]，minw=band[0]，此时minw=80,totalp=0，继续循环:s=1<=5，满足条件，minp=32767,继续：t=1<=d[1]=3；满足条件进入判断语句：a[1][1].bw=100>minw=80，a[1][1].price=30<minp=32767，满足条件，

minp=30

此时程序第一次执行到76行；

之后s=1的情况下，a[1][2].price=40>minp=30，不执行76行语句。

S++，s=2<=5，满足循环条件，此时a[2][1].bw=80>=minw=80且a[2][1].price=40<minp=32767,执行76行后，

minp=40

1. 写出样例输入时程序的执行结果。（10分）

0.500

本题评分标准：

1. merge\_sort执行过程分析：
2. 分析方法正确。（10分）
3. 分析结果正确。（10分）
4. 分析过程完整，准确。（10分）
5. Merge时间复杂度分析：
6. 分析方法正确。（10分）
7. 计算表达式正确，化简方法正确，结果正确。（5分）
8. merge\_sort时间复杂度分析：
9. 分析方法正确。（10分）
10. 计算表达式正确，化简方法正确，结果正确（5分）
11. minp的值正确（5分），分析方法正确（10分），分析中间结果正确（5分）。

2．阅读下面的程序，按要求回答下面的问题。

1. #include<cstdio>
2. #include<cstring>
3. #include<iostream>
4. using namespace std;
5. char chess[10][10];
6. int sign[10];
7. int n, k, ans;
8. void dfs(int x, int k)
9. {
10. if (k == 0) {
11. ans++;
12. return;
13. }
14. if (x+k-1 > n) {
15. return;
16. }
17. for (int i = x; i <= n; i++) {
18. for (int j = 1; j <= n; j++) {
19. if (!sign[j] && chess[i][j] == '#') {
20. sign[j] = 1;
21. dfs(i + 1, k - 1);
22. sign[j] = 0;
23. }
24. }
25. }
26. }
27. int main() {
28. while (cin >> n >> k) {
29. memset(chess, 0, sizeof(chess));
30. memset(sign, 0, sizeof(sign));
31. if (n == -1 || k == -1) {
32. break;
33. }
34. for (int i = 1; i <= n; i++) {
35. for (int j = 1; j <= n; j++) {
36. cin >> chess[i][j];
37. }
38. }
39. ans = 0;
40. dfs(1, k);
41. cout << ans << endl;
42. }
43. return 0;
44. }

样例输入：

5 5

#.#.#

..###

##.#.

#.##.

.#.#.

-1 -1

要求：

1. 写出样例输入时程序运行到第39行时变量n，k和数组chess[n+1][n+1]中元素的值。（10分，每错1个扣1分，直到本小题分数扣完为止）

n = 5， k = 5,

chess[0][0] = ‘\000’ chess[1][0] = ‘\000’ chess[2][0] = ‘\000’

chess[0][1] = ‘\000’ chess[1][1] = ‘#’ chess[2][1] = ‘.’

chess[0][2] = ‘\000’ chess[1][2] = ‘.’ chess[2][2] = ‘.’

chess[0][3] = ‘\000’ chess[1][3] = ‘#’ chess[2][3] = ‘#’

chess[0][4] = ‘\000’ chess[1][4] = ‘.’ chess[2][4] = ‘#’

chess[0][5] = ‘\000’ chess[1][5] = ‘#’ chess[2][5] = ‘#’

chess[3][0] = ‘\000’ chess[4][0] = ‘\000’ chess[5][0] = ‘\000’

chess[3][1] = ‘#’ chess[4][1] = ‘#’ chess[5][1] = ‘.’

chess[3][2] = ‘#’ chess[4][2] = ‘.’ chess[5][2] = ‘#’

chess[3][3] = ‘.’ chess[4][3] = ‘#’ chess[5][3] = ‘.’

chess[3][4] = ‘#’ chess[4][4] = ‘#’ chess[5][4] = ‘#’

chess[3][5] = ‘.’ chess[4][5] = ‘.’ chess[5][5] = ‘.’

1. 分析样例输入时函数dfs的执行过程。（30分）

该算法为寻找一个二维数组中不同行不同列的相同符号的组数。

在输入样例中，n=5,因此创建了一个5\*5的二维数组，其中符号为：

#.#.#

..###

##.#.

#.##.

.#.#.

又，k=5，即寻找该数组中在不同行不同列的5个为一组的“#”有多少组。算法主体为dfs函数，首先由第一行开始，依次首先执行dfs（1，5），在函数体内，需判断标志数组sign[]是否为0，以及该行该列字符是否为“#”，易知第一行第一列符合，接着递归调用，即执行dfs（2，4），即开始判断第二行是否有符合的字符，此时只需再另寻4个“#”即可，因此k=k-1，此时判断sign[j]时，因为上文已经把sign[1]置1，也即第一列存在“#”号，所以进入sign[2]判断，即判断第二列是否有符合的字符，由样例可知，第二列存在符合条件的字符，所以k=k-1，再次递归时进入k==0判断，数量ans+1。后续也即按照上述流程进行，最后得出ans=9，也即该二维数组中共有9组个数为5，字符为“#”的字符对。

1. 分析函数dfs的时间复杂度，写出分析过程。（30分）

当chess数组全为‘#’时，时间复杂度最大，考虑第一个字符‘#’在第一行时，有n种情况，则第二个字符需在（n-1）\*（n-1）数组中，推广可知，剩下n-1个字符的话需在（n-1）\*（n-1）数组中；

dfs函数在执行前需判断k的大小，时间复杂度\*1，设关于时间复杂度的函数为T（n），dfs（1，n）的时间复杂度为T（n），由分析可知，有如下递归关系

T(n)=n\*T(n-1)+1

又，T（1）=1，化简得

T(n)=n!

即该算法时间复杂度为O(n!).

1. 写出样例输入时函数dfs执行结束之后数组sign中各个元素的值(每错1个扣3分，直到本小题分数扣完为止)。（20分）

sign:[0] = 0,

1. = 0,
2. = 0,
3. = 0,
4. = 0,
5. = 0,
6. = 0,
7. = 0,
8. = 0,
9. = 0
10. 写出样例输入时整个程序执行结束之后变量ans的值。（10分）

ans = 9,

本题评分标准：

1. dfs执行过程分析：（30分）
2. 分析方法正确。（10分）
3. 分析结果正确。（10分）
4. 分析过程完整，准确。（10分）
5. dfs时间复杂度分析：（30分）
6. 分析方法正确。（15分）
7. 计算表达式正确，化简方法正确，结果正确。（15分）

提交内容：Word文档，文档中包含所有的问题的答案，图要求用Visio绘制。

提交截止时间：2022年4月9日20：00，每延后一天扣20分。

提交方式：电子稿，命名规则：学号\_姓名\_作业2.rar

邮件发送给：hyhuang@home.swjtu.edu.cn