题目要求：<23-P2392-kkksc03考前临时抱佛脚-题目要求.docx>

## 一、总体分析：

四个科目其实就是四组数据。考虑一个科目单独分析。

显然若当前要处理的问题的时间和为t，最佳答案是t/2，也就是两边脑所耗时间差为0。

于是现在要求的问题就是**使得两边脑所耗时间差尽量小**。

如果只考虑一边脑耗时，由于两边脑是对称的，我们假设这个耗时肯定比t/2小。那么我们要使这个耗时尽量接近t/2。

然后又想到了背包。保证比t/2小，又要使取得的价值尽量大，可以转化为一个费用与价值相同的背包问题。背包大小为t/2，一个题目的费用和价值相等，都是所耗的时间。

**假设背包求得的最大价值为v,显然另一个脑耗时为t-v，则总耗时为max(v,t-v)。**

## 二、非正解——贪心

既然是算较短的时间，如果左脑所用时间少就加在左脑，如果右脑所用时间少就加在右脑，然而并不是正解

1. #include<bits/stdc++.h>
2. using namespace std;
3. int a[5],i,j,sum1,sum2,t,homework;
4. int main(){
5. for(i=1;i<=4;i++)
6. cin>>a[i];*//输入*
7. for(i=1;i<=4;i++){
8. sum1=sum2=0;*//两边脑子时间清零*
9. for(j=1;j<=a[i];j++)
10. {cin>>homework;
11. if(sum1<=sum2) sum1+=homework;
12. else sum2+=homework;}*//哪边时间短就加在哪边*
13. t+=max(sum1,sum2);*//取较长时间累加*
14. }cout<<t;*//输出*
15. return 0;
16. }

## 三、背包

对于一道题只有两个状态，一是加到左脑，二是加到右脑，所以是01背包，这里还可以用另一个思想，将一边的脑子加到（从左侧）最接近一半则另一边脑子时间就是正解

1. #include<bits/stdc++.h>
2. using namespace std;
3. int a[5],i,j,k,sum,t,homework[21],dp[2501];
4. int main(){
5. for(i=1;i<=4;i++)
6. cin>>a[i];
7. for(i=1;i<=4;i++){
8. sum=0;
9. for(j=1;j<=a[i];j++)
10. {cin>>homework[j];*//输入*
11. sum+=homework[j];}*//总时间累加*
12. for(j=1;j<=a[i];j++)
13. for(k=sum/2;k>=homework[j];k--)*//只要是总和的一半*
14. dp[k]=max(dp[k],dp[k-homework[j]]+homework[j]);*//01背包*
15. *//此处进行了优化避免了不必要的循环次数，也相当于从规定的左侧方向接近一半*
16. t+=sum-dp[sum/2];*//累加为另一个脑子*
17. for(j=1;j<=sum/2;j++)
18. dp[j]=0;*//清零*
19. }
20. cout<<t;*//输出*
21. return 0;
22. }

## 四、模板总结

**典型的01背包问题**

几乎是01背包模板题，无脑刷过

**不同点**

要分四次记录答案

**坑**

记牢每次一定要初始化背包数组

## 五、简洁地写代码

首先理清思路:

* 1.所求最少时间即4科最少时间的总和
* 2.因为kkksc03是双核,故单科最少总时间分配给两核的时间应尽量相同
* 3.故:以单科总时间一半为界限,做01背包是所花时间尽量靠近与HALF
* 4.背包结束后 dp[HALF]总是小于等于ALL-dp[HALF],应累加较大值即后式

## 六、相似的题：

小S新买了一个掌上游戏机，这个游戏机由两节5号电池供电。为了保证能够长时间玩游戏，他买了很多5号电池，这些电池的生产商不同，质量也有差异，因而使用寿命也有所不同，有的能使用5个小时，有的可能就只能使用3个小时。显然如果他只有两个电池一个能用5小时一个能用3小时，那么他只能玩3个小时的游戏，有一个电池剩下的电量无法使用，但是如果他有更多的电池，就可以更加充分地利用它们，比如他有三个电池分别能用3、3、5小时，他可以先使用两节能用3个小时的电池，使用半个小时后再把其中一个换成能使用5个小时的电池，两个半小时后再把剩下的一节电池换成刚才换下的电池（那个电池还能用2.5个小时），这样总共就可以使用5.5个小时，没有一点浪费。 （noi.openjudge.cn 4.6贪心 2469）

没想到这个人不能分开做题目，开始一道就必须做完

因此两道题的区别是：临时抱佛脚必须“锲而不舍”，玩游戏可以“半途而废”

**电池问题的求解：**

1、如果一根电池就比其他所有电池时间和都长，那么最短时间就是这最长的电池持续的时间

2、如果这根电池时间比其他总和短，那么最短时间就是所有持续时间总和/2（可以证明）

## 七、我的代码：

附上文件路径：<23-P2392-kkksc03考前临时抱佛脚.cpp>

1. #include<stdio.h>
2. #define N 4
3. #define max(m,n) (((m)>(n))?(m):(n))
4. int main(void)
5. {
6. int work[5],sum=0,dp[100][1500]={0},ans=0;
7. for(int i=0;i<N;i++)scanf("%d",&work[i]);
8. int time[5][61];
9. for(int i=0;i<N;i++)
10. {
11. sum=0;
12. for(int j=0;j<work[i];j++)
13. {
14. scanf("%d",&time[i][j]);
15. sum+=time[i][j];
16. }*//读入该科目每道题所用的时间并计算所有题用时和*
17. if(work[i]==1) ans+=time[i][0];*//如果仅有一道题，答案易得*
18. else if(work[i]==2) ans+=max(time[i][0],time[i][1]);
19. *//如果仅有两道题，答案即为较大的时间值*
20. else
21. {*//否则进入背包算法，物品数是work[i]，物品的价值和用时（近似于体积）相同。*
22. for (int p=1;p<=work[i];++p)*//work[i]指示该科目题目的数目*
23. {
24. for (int q=1;q<=sum/2;++q)
25. { *//背包容量也即总时间和的一半*
26. if(q>=time[i][p-1])*//q表示当前的背包容量*
27. *//如果q大于等于当前题目所需的时间那么物品数为p的背包问题归结于*
28. *//放入p和不放入p两种情况，取最大值*
29. dp[p][q]=max(dp[p-1][q],dp[p-1][q-time[i][p-1]]+time[i][p-1]);
30. else*//如果q小于当前题目所需的时间，则p无法放入结果唯一*
31. dp[p][q]=dp[p-1][q];
32. }
33. }
34. ans+=max(sum-dp[work[i]][sum/2],dp[work[i]][sum/2]);
35. *//该科目总用时即为两半大脑用时的最大值*
36. }
37. *//  printf("--%d--\n",ans);*
38. }
39. printf("%d",ans);
40. }