实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 实验名称 | **实验三 贪心算法设计** | | |
| 实验日期 | 2023年4月11日 |  |  |
| 学 号 | 2021213193 | 姓 名 | 李田 |
| 专业班级 | 2021级软件工程 2班 | | |
| 指导教师 | 谷志新 | | |

东北林业大学

软件工程专业

|  |
| --- |
| 1. 实验目的   掌握贪心算法的基本思想，掌握贪心算法的设计步骤及算法实现。 |
| 1. 实验环境   Windows 7 以上操作系统，PC机，codeblocks环境 |
| 1. 实验内容及结果   **1、最优服务次序问题**  （1）问题描述：  　　设有n 个顾客同时等待一项服务。顾客i需要的服务时间为ti, 1<=i <= n 。应如何安排n个顾客的服务次序才能使平均等待时间达到最小?平均等待时间是n 个顾客等待服务时间的总和除以n。  （2）编程任务：  　　对于给定的n个顾客需要的服务时间，编程计算最优服务次序。  （3）数据输入：  　　第一行是正整数n，表示有n 个顾客。接下来的1行中，有n个正整数，表示n个顾客需要的服务时间。  （4）结果输出:  　　计算出的最小平均等待时间。  （5）输入示例  10  56 12 1 99 1000 234 33 55 99 812  （6）输出示例  532.00  #include <bits/stdc++.h>  using namespace std;  typedef long long LL;  #define PLL pair<LL, LL>  #define PII pair<int, int>  #define rep(i, a, b) for (int i = a; i <= b; i++)  #define dep(i, a, b) for (int i = a; i >= b; i--)  #define lowbit(x) (x & -x)  #define x first  #define y second  const double eps = 1e-8;  clock\_t startTime;  #define l first  #define r second  double getCurrentTime()  {    return (double)(clock() - startTime) / CLOCKS\_PER\_SEC;  }  const int N=2e5+10;  int a[N],pre[N];  bool cmp(int a,int b)  {    return a<b;  }  void solve()  {     int n;     cin>>n;     rep(i,1,n)cin>>a[i];     sort(a+1,a+1+n,cmp);     LL ans=0;     rep(i,1,n)     {       ans+=(n-i+1)\*a[i];     }     cout<<(double)ans/n<<endl;    }  int main()  {    int tt;    tt = 1;    startTime = clock();    while (tt--)    {      solve();      // printf("%.2lf",getCurrentTime());    }  }  **测试样例：**    **2、区间相交问题**  （1）问题描述：  　　给定x 轴上n 个闭区间。去掉尽可能少的闭区间，使剩下的闭区间都不相交。  （2）编程任务：  　　给定n 个闭区间，编程计算去掉的最少闭区间数。  （3）数据输入：  　　第一行是正整数n，表示闭区间数。接下来的n行中，每行有2 个整数，分别表示闭区间的2个端点。  （4）结果输出:  计算出的去掉的最少闭区间数。  （5）输入示例  3  10 20  10 15  20 15  #include <bits/stdc++.h>  using namespace std;  typedef long long LL;  #define PLL pair<LL, LL>  #define PII pair<int, int>  #define rep(i, a, b) for (int i = a; i <= b; i++)  #define dep(i, a, b) for (int i = a; i >= b; i--)  #define lowbit(x) (x & -x)  #define x first  #define y second  const double eps = 1e-8;  clock\_t startTime;  #define l first  #define r second  double getCurrentTime()  {    return (double)(clock() - startTime) / CLOCKS\_PER\_SEC;  }  const int N=2e5+10;  struct seg{      int l,r;  }segs[N];  bool cmp(seg a,seg b)  {     if(a.l!=b.l) return a.l<b.l;     return a.r<b.r;    }  void solve()  {      int n;      cin>>n;      rep(i,1,n) {          int l,r;          cin>>l>>r;          if(l>r) swap(l,r);          segs[i].l=l;          segs[i].r=r;      }      sort(segs+1,segs+1+n,cmp);      int tl=segs[1].l,tr=segs[1].r;      int ans=0;      rep(i,2,n)      {         if(segs[i].l<=tr) ans++;         tr=max(tr,segs[i].r);      }      cout<<ans<<endl;    }  int main()  {    int tt;    tt = 1;    startTime = clock();    while (tt--)    {      solve();      // printf("%.2lf",getCurrentTime());    }  }  测试样例：    **3、汽车加油问题**  问题描述：一辆汽车加满油后可行驶n km。旅途中有若干加油站。设计一个有效算法，指出应在哪些加油站停靠加油，使沿途加油次数最少。  算法设计：对于给定的n和k个加油站位置，计算最少加油次数。  数据输入：n：表示汽车加满油后可行驶nkm  k：旅途中有k个加油站  k+1个整数：表示第k个加油站与第k-1个加油站之间的距离。第0个加油站表示出发地，汽车已加满油。第k+1个加油站表示目的地。  数据输出：最少加油次数和具体在哪几个加油站加油。  例如： n=7 k=7  K+1个整数：1 2 3 4 5 1 6 6  最优值：4  #include <bits/stdc++.h>  using namespace std;  typedef long long LL;  #define PLL pair<LL, LL>  #define PII pair<int, int>  #define rep(i, a, b) for (int i = a; i <= b; i++)  #define dep(i, a, b) for (int i = a; i >= b; i--)  #define lowbit(x) (x & -x)  #define x first  #define y second  const double eps = 1e-8;  int main()  {      int m,n;      scanf("%d%d",&m,&n);      int i,j;      int a[n+1];      for(i=0;i<=n;i++)      {          scanf("%d",&a[i]);      }      int sum=a[0];//从第一站记起      int count=0;      for(i=0;i<=n;i++)      {          if((sum+a[i])<=m)          {              sum+=a[i];          }          else if(a[i]>m)          {              printf("No Solution!");              return 0;          }          else          {              count++;              sum=a[i];//更新距离          }      }      printf("%d\n",count);      return 0;  }  测试样例： |
|  |

|  |
| --- |
| 四、实验过程分析与讨论  在解决区间相交问题时由于没有考虑到数据的特殊性，导致当区间左端点大于右端点时，得出错误的答案：  更正代码：  rep(i,1,n) {          int l,r;          cin>>l>>r;          if(l>r) swap(l,r);          segs[i].l=l;          segs[i].r=r;      } |
| 五、指导教师意见  指导教师签字：  年 月 日 |
|  |