**《数据结构综合设计》实验报告**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 学院 | 人工智能与大数据学院 | 专业 | 虚拟现实技术 | 班级 | 21级3班 | 学生姓名 | 杜晓萌 |
| 实验  周次 | 13-14 | 实验  日期 | 2023.5.25 | 学时 | 6 | 教师姓名 | 李昊康 |
| 项目名称 | | 排序的应用 | | | | | |
| 实验  类别 | 🗹验证型实验 🞎设计型实验 🞎综合型实验 🞎其它 | | | | | 成绩：94 | |
| 1. 实验目的及具体要求   实验目的：  1.实现多种类型的排序算法（插入排序、交换排序、选择排序、归并排序等）；  2.理解排序过程；  3.计算比较次数和移动次数，对比分析算法性能的优劣与适用场景；  具体要求：  编写程序实现插入排序、希尔排序、冒泡排序、快速排序、简单选择排序、堆排序。中任意3种。   1. 实验仪器、设备和材料   硬设备：PC机  软件环境：Windows VS2019   1. 实验内容、步骤及实验数据记录   void Inserte\_sort(int A[], int n)    {    int i, j, temp,a=0,aa=0;    for (i = 1, a++; i < n; i++, a++)    if (A[i] < A[i - 1])    {    temp = A[i];    for (a++, aa++, j = i - 1; j >= 0 && A[j] > temp; j--, a++, aa++)    A[j + 1] = A[j];    A[j + 1] = temp;    }  }    void Select\_sort(int A[], int n)  {    int i, j, k,a=0,aa=0;    for (i = 0, a++; i < n - 1; i++, a++)    {    k = i;    for (j = i + 1, a++; j < n; j++, a++)    if (A[j] < A[k])    {    k = j; a++;    }    if (k != i)    {    aa += Change\_a\_b(&A[i], &A[k]);    }    }  }    void Bubble\_sort(int A[], int n)    {    int i, j, a=0;    long long aa=0;    for (i = n - 1; i >= 0; i--)    {    for (j = 0; j < i; j++)    {    if (A[j] > A[j + 1])    aa += Change\_a\_b(&A[j], &A[j + 1]);    }    }    a = (n - 1) \* n/ 2;    }    void Quick\_sort(int A[], int L, int R)    {    int i = L, j = R, T = A[L];    if (L > R) return;    while (i != j)    {    while (A[j] >= T && i < j) { j--;c++; }    while (A[i] <= T && i < j) { i++; c++; }    if (i < j)cc += Change\_a\_b(&A[i], &A[j]);    }    if (L != i)    cc += Change\_a\_b(&A[L], &A[i]);    Quick\_sort(A, L, i - 1);    Quick\_sort(A, i + 1, R);    }    void Shell\_sort(int A[], int n)    {    int i, j, k,a=0,aa=0;    for (k = n >> 1, a++; k > 0; k = k >> 1, a++)  {    for (i = k, a++; i < n; i++, a++)    {    for (j = i - k, a++; j >= 0; j -= k, a++)    if (A[j] > A[j + k])    aa += Change\_a\_b(&A[j], &A[j + k]);    }    }    }    int Merge(int S[], int L, int M, int R, int T[])    {    int i = M, j = R, k = 0, count = 0;    while (i >= L && j > M)    {    b++;    if (S[i] > S[j])    {    T[k++] = S[i--];    }    else T[k++] = S[j--];    count++;    }    while (i >= L) { b++; T[k++] = S[i--]; }    while (j > M) { b++; T[k++] = S[j--]; }    for (i = 0, b += k; i < k; i++) S[R - i] = T[i];    return count;    }    void Merging\_sort(int S[], int L, int R, int T[])    {    int M;    if (L < R)    {    M = (L + R) >> 1;  Merging\_sort(S, L, M, T);    Merging\_sort(S, M + 1, R, T);    bb += Merge(S, L, M, R, T);    }    }    18515164  分析：  冒泡：好实现，速度不慢，使用于轻量级的数据排序。  插入排序：也使用于小数据的排序，但是我从他的思想中学到怎么插入一个数据。  选择排序：我学会了怎么去获得最大值，最小值等方法。  合并排序：我学会分而治之的方法，而且在合并两个数组的时候很适用。 | | | | | | | |

说明：1. 实验周次：填写实际上课周，如第5-8周上课填“5-8”或第10周上课填“10”。

1. 实验报告各部分内容需详实填写，按实验指导书上的评分标准给出分数。
2. 实验类型参考实验类型说明文件。