به نام خدا

|  |  |  |  |
| --- | --- | --- | --- |
| مدرس: مظفر بگ محمدي | دانشگاه ايلام | ترم اول سال تحصيلي 98-97 | سري ششم تمرينات ساختمان داده |

1. آرایه‌ی نامرتب 4, 11, 3, 2, 18, 25 14, 3 را با توجه به کدهای زیر مرتب کنید. تمام مراحل کار را نشان دهید.

private void quicksort(int low, int high) {

int i = low, j = high;

int pivot = numbers[low + (high-low)/2];

while (i <= j) {

while (numbers[i] < pivot) { i++; }

while (numbers[j] > pivot) { j--; }

if (i <= j) { exchange(i, j); i++; j--; }

}

if (low < j) quicksort(low, j);

if (i < high) quicksort(i, high);

}

private void exchange(int i, int j) {

int temp = numbers[i];

numbers[i] = numbers[j];

numbers[j] = temp;

}

public class Mergesort {

private int[] numbers; private int[] helper;

private void mergesort(int low, int high) {

if (low < high) {

int middle = low + (high - low) / 2;

mergesort(low, middle);

mergesort(middle + 1, high);

merge(low, middle, high);

}

}

private void merge(int low, int middle, int high) {

for (int i = low; i <= high; i++) {

helper[i] = numbers[i];

}

int i = low; int j = middle + 1; int k = low;

while (i <= middle && j <= high) {

if (helper[i] <= helper[j]) { numbers[k] = helper[i]; i++;}

else { numbers[k] = helper[j]; j++; }

k++;

}

while (i <= middle) { numbers[k] = helper[i]; k++; i++; }

}

}

public static void InsertionSort( int [ ] num)

{

int j; int key; int i;

for (j = 1; j < num.length; j++)

{

key = num[ j ];

for(i = j - 1; (i >= 0) && (num[ i ] < key); i--){

num[ i+1 ] = num[ i ];

}

num[ i+1 ] = key;

}

}

1. تابعی بنویسید که تعداد نودهای غیر انتهایی یک درخت دودویی را محاسبه کند.
2. متدهای مرتب سازی درجی، ادغام و سریع را پیاده کنید.
   1. در تابع main یک میلیون عدد تصادفی تولید نمایید و در یک آرایه ذخیره‌کنید. نیمه‌ی پایین آرایه را با مرتب سازی درجی و نیمه‌ی بالا را با مرتب‌سازی سریع مرتب کنید. سپس آرایه‌های مرتب شده را به روش مرتب سازی درجی با هم ادغام کنید.
   2. دوباره یک آرایه تصادفی تولید و کل آرایه را به روش مرتب سازی درجی مرتب کنید.
   3. دوباره یک آرایه تصادفی تولید و کل آرایه را به روش مرتب سازی ادغام مرتب کنید.
   4. دوباره یک آرایه تصادفی تولید و کل آرایه را به روش مرتب‌سازی سریع مرتب کنید.
   5. زمان مراحل قبل را با هم مقایسه کنید.
3. مرتب سازی سطلی را پیاده و عملکرد آنرا روی اعداد ۶ رقمی امتحان کنید.
4. یک متد بنویسید که تعداد نودهای غیر برگ یک درخت دودویی را بشمارد.
5. یک متد بنویسید که تمام نودهای مابین a و b (0<b<a) را حذف کند.
6. درخت جستجوی دودویی را برای اعداد طبیعی پیاده کنید. سپس در یک حلقه ۵۰ عدد تصادفی بین ۱ تا ۱۰۰ را وارد درخت کنید. پیمایش به ترتیب درخت را نمایش دهید. سپس به ترتیب نودهای ۱۲، ۶۲، ۸، ۹۲ را در صورت وجود حذف کنید و پیمایش به ترتیب درخت را دوباره نمایش دهید.
7. درخت جستجوی دودویی را برای کاراکترها پیاده کنید. یک رشته از کاربر بگیرید و کاراکترهای آنرا از چپ به راست وارد درخت کنید. تمام پیمایشهای درخت را اجرا کنید.
8. یک عبارت میانوندی پرانتزبندی شده را از کاربر دریافت کنید و آنرا وارد درخت دودویی کنید. سپس نمایشهای پسوندی و پیشوندی درخت را اجرا کنید.
9. توده ی حداقل را با آرایه برای اعداد طبیعی پیاده کنید. سپس اعمال زیر را به ترتیب از چپ به راست انجام دهید.

add(4), add(18), add(11), add(2), add(31), add(42), delete(), delete(), add(13), add(5), add(1), delete()