3 giờ chieu, 29/3

thêm các file txt thuật toán còn thiếu

sửa đoạn code trong load Thuật toán

sửa code nút bắt đầu

đổi màu ô vuông cuối cho thuật toán interchange sort

sửa code selection sort

sửa code nút làm lại (-> sửa code nút nạp, nạp từ file)

9 giờ tối, 29/3

thêm thanh tốc độ:

hàm mới:

+changeSpeed() //trong thao tác tab auto

+các hàm liên quan đến thanh tốc độ //trong thao tác tab auto

0h'21 30/3

-Thêm thuật toán Bubble sort

11h55 30/3

sửa code bubble sort

2h chiều 31/3

thêm hàm thread.sleep vào các hàm thuật toán

thêm hàm refresh trong các hàm thuật toán để update màu ô vuông

sửa vài lỗi khác

23h 31/3

Sửa interchange và selection sort

9h 4/1

Thêm label vị trí I, j

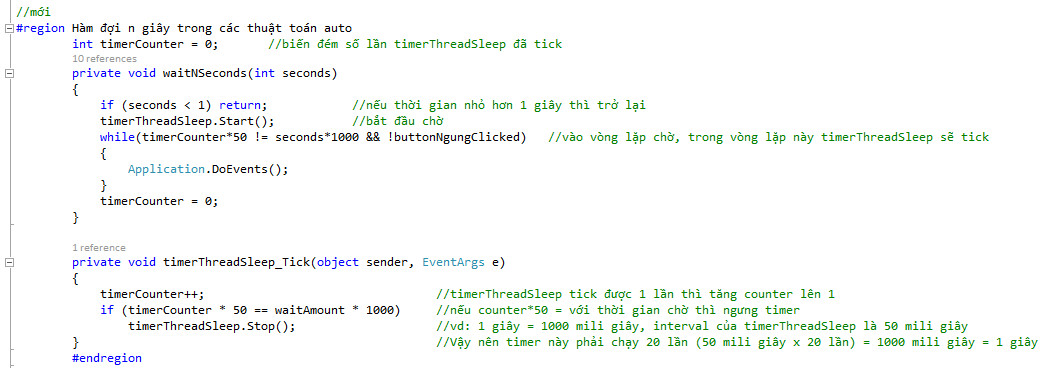
2h16 2/4

Thêm thuật toán Insertion Sort – chưa test bug

1h57 3/4

Thêm thuật toán Binary Insertion Sort – chưa test bug

4/3

Thay đổi hàm thread.sleep thành waitNSeconds

Hàm waitNSeconds(int seconds) dùng để chờ n giây r mới chạy tiếp, mặc định 5 (0,5 giây) giây khi load form (nếu là 1 giây thì seconds = 10)  
Hàm sử dụng timerThreadSleep để đếm

Hoạt động:

1. Start timerThreadSleep

2. Vào vòng lặp while DoEvent để thực hiện sự kiện tick

3. Mỗi lần tick, biến đếm số lần tick timerCounter tăng lên 1, tăng đến khi gặp điểu kiện trong hình thì dừng. Vì timerThreadSleep có interval = 50 miligiây (càng nhỏ thì càng dễ ấn nút ngưng) nên biến đếm phải đếm 20 lần cho 1 giây (20 lần \* 50 miligiây = 1000 miligiay = 1 giây), 2 giây thì 40 lần

4. Đếm đủ số giây thì dừng hoặc gặp sự kiện Ngưng thì dừng timer

Thêm liên quan:

+Dừng timerThreadSleep trong nút ngưng

+Set 2 cờ exitFlag = false trong nút tiếp tục

+Biến waitAmount dùng để ngưng n giây nằm trong region sự kiện thanh tốc độ, chuyển mức tốc độ thì waitAmount cũng đổi theo (Mức 4 = 1s, 3 = 2s,..)

+waitAmount = 0.5 (seconds = 5) giây khi load form

5/4

sửa lỗi Interchange, Bubble, Selection sort, Insertion

Binary Insertion đang sửa

Hàm đổi chổ 2 ô vuông có thêm 2 tham số i và j, hàm sẽ đổi chổ 2 ô vuông nào có vị trí là i, j (ô vuông i đi từ phải qua trái, ô vuông j đi từ trái qua phải)

6/4

Heap sort

8/4

Highlight interchange sort

thêm shaker sort, chỉnh code Heap sort, chỉnh focus các nút

10/4 thêm hàm nút tiến tab step by step

12/4

thêm các hàm quan trọng:

//hàm này chỉ thực hiện khi đã hoàn thành sắp xếp

private void enableControlsAfterFinish()

{

isStepbyStepStarted = false; //set lại bằng false cho lần thực hiện mới tiếp

ButtonTien.Enabled = false; //đã hoàn thành bước cuối cùng thì ko thể tiến thêm đc nữa

buttonBatDau.Enabled = false; //ko cho phép bắt đầu với dãy số đã hoàn thành

buttonNgung.Enabled = false; //ko cho phép ngưng khi chưa bắt đầu

//cho phép nạp dãy số, chọn thuật toán, chọn thứ tự khi hoàn thành

ButtonNap.Enabled = true;

ButtonRandom.Enabled = true;

fileButton.Enabled = true;

ComboBoxThuatToan.Enabled = true;

RadioButtonGiamDan.Enabled = true;

RadioButtonTangDan.Enabled = true;

ButtonLamLai.Enabled = true;

}

Câu if thay thế dùng để dừng đợi nút ngưng/tiếp tục/tiến/làm lại:

if (isStepbyStepStarted) //nếu đang làm step by step thì đợi nút tiến

{

try

{

await waitForbuttonTienClick(); //dừng để đợi người dùng ấn nút Tiến để tiếp tục

}

catch

{

}

if (buttonLamLaiClicked) //nếu button làm lại đc ấn thì thoát hẳn thuật toán

{

buttonLamLaiClicked = false;

return;

}

continueFlag = new CancellationTokenSource(); //tạo lại token (lượt dừng) mới cho lần dừng tiếp theo

}

else

{

//nếu ko phải đang chạy step by step thì là đang chạy auto

waitNSeconds(waitAmount);

if (buttonNgungClicked) //nếu đang đợi n giây mà nút ngưng đc ấn thì gọi hàm đợi nút tiếp tục

{

try

{

await waitForTiepTucClick(); //chờ nút tiếp tục được bấm

}

catch

{

}

if (buttonLamLaiClicked) //nếu button làm lại đc ấn thì thoát hẳn thuật toán

{

buttonLamLaiClicked = false;

return;

}

continueAutoFlag = new CancellationTokenSource(); //nếu nút ngưng đc bấm và sau đó nút tiếp tục đc ấn thì phải tạo lại token

}

}